ANALYSIS AND EFFICIENT SIMULATION OF QUEUEING MODELS OF TELECOMMUNICATION SYSTEMS

Pieter Tjerk de Boer

ISBN 90–365–1505–X
ISSN 1381–3617, CTIT Ph.D.-thesis series no. 00–01

Centre for Telematics and Information Technology,
P.O. Box 217, 7500 AE Enschede, The Netherlands

ANALYSIS AND EFFICIENT SIMULATION OF QUEUEING MODELS OF TELECOMMUNICATION SYSTEMS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 19 oktober 2000 te 15.00 uur.

door

Pieter Tjerk de Boer

geboren op 18 januari 1972

te Wildervank (gem. Veendam)

Dit proefschrift is goedgekeurd door de promotor en de assistent-promotor,

prof. dr. ir. I.G.M.M. Niemegeers
dr. ir. V.F. Nicola

Contents

Chapter 1

Introduction

This thesis is about analysis and efficient simulation of queueing models of telecommunications systems. In this introductory chapter, some background and motivation for this work is provided, and an outline of the thesis' content presented.

1.1 Packet-switched telecommunication systems

The first electric and electronic telecommunications systems, such as the telegraph and telephone networks, were based on a concept known as *circuit-switching*. In such a system, a separate "circuit" (e.g., a pair of wires or a slice of radio spectrum) is reserved for each connection, and remains reserved until that connection is no longer needed. This is a straightforward concept that fits naturally to the notion of a (telephone) conversation, and that can be implemented on the basis of relatively simple technology, such as mechanical switches and relays; see [vHK68] for an overview of such technologies. However, circuit switching is not very efficient: in a typical telephone conversation only one of the participants is speaking at a time, while the other is listening, so only half of the two-way channel is actually used. Furthermore, modern computer telecommunication needs (such as e-mail and file transfer) are more "bursty" in nature: they can use a large bandwidth but need it only for a short time, while circuit switching would assign a relatively small bandwidth for a long time.

Starting in the 1960s, a different type of communication network was developed: the ARPANET, which evolved into the Internet (based largely on IP, the Internet Protocol) known today. In the 1990s, the ATM (Asynchronous Transfer Mode) network type was developed [Onv94]. These networks are based on the concept of *packet-switching*. In a packet- (or cell-, in the case of ATM) switched network, no wires or radio or cable bandwidth is reserved for a connection. Instead, the network offers a packet-delivery service: information to be transmitted is offered to the network in the form of packets, each containing some limited number of characters (bytes) of information, as well as some additional information (the "header") which tells the network to what destination this packet must be transported. Long messages can be split over several packets, and packets are also used to acknowledge the correct receipt of information. Clearly, in such a system no resources will be wasted for idle connections, since an idle connection will not inject any packets into the network; thus, all available transmission capacity of the network links can be used for active connections. However, in such a network it can easily happen that while a link is busy transmitting one packet, another packet arrives which needs to be transmitted over the same link. In that case, the newly arriving packet needs to be stored in a buffer memory, from which it is read and transmitted when the link is available again. If the buffer memory is already totally filled up with packets, the newly arriving packet needs to be dropped. Thus, we see that the advantage of more efficient utilization of the network links comes at a cost: the buffering introduces an uncertain amount of delay, and a risk that packets are never delivered due to buffer overflow.

The consequences of varying delays and non-delivery depend very much on the application the network is used for. Originally, packet-switched networks were intended for data-communication; e.g., the transfer of e-mail messages and files from one computer to another. A variable delay hardly hurts such an application: it just takes a bit longer to transport the message. Packet loss would hurt very much, if it would mean that part of an e-mail would simply disappear. However, protocols (such as TCP, the Transmission Control Protocol in the Internet) are used which check whether all packets have arrived and, if necessary, make sure a missing packet is retransmitted until it has finally arrived. Of course this takes time and additional bandwidth, so the transport delay increases.

If one tries to use a packet-switched network for applications with more real-time demands, such as telephony or a video stream, the variable delay caused by the buffering can present a problem: if a packet of data arrives too late, it is no longer useful. Also packet loss presents a problem, since a retransmission is usually too time-consuming. If one packet of data (containing a fraction of a second worth of telephone sound, for example) is lost or arrives too late, some kind of interruption in the playback will occur. If this happens rarely, it is not really a problem, but if it happens often it can be very disturbing for the usability of the connection.

Packet loss and delay and their impact on the performance of the communication are known as *Quality of Service* (QoS) considerations. A major issue in the design of modern packet-switched telecommunication systems is the prediction of the QoS that the network's user will get; or, conversely, how to engineer the network such that the service offered will satisfy the user's quality requirements while maintaining efficient usage of network resources.

1.2 Modelling and analysis

In order to mathematically investigate the packet loss and delay properties of a telecommunication network or a component of such a network, a model of the system must be formulated in which such phenomena are expressed. This can be done using a branch of mathematics known as *queueing theory*. In queueing theory, models are studied of systems in which "customers" (randomly) arrive at a "service station" in order to be "served"; since there may be other customers ahead of them, they may need to wait in a buffer or "queue". Queueing models are characterized by the probability distribution of the time between arrivals, the probability distribution of the time needed to serve a customer, size of the buffer space (if finite), queueing policy (e.g., first come first served), etc. One of the original motivating applications for the development of queueing theory was in fact the telephone network at the beginning of the 20th century: the waiting time until a call could be handled by an operator, and the probability of a call blocking due to the unavailability of free lines were among the first results by A. K. Erlang [BHJ60].

Already at the start of the development of packet-switched telecommunication networks in the 1960s, queueing theory was applied to models of such networks [Kle64]. A queueing model of a communication network typically includes one or more sources which send packets (cf. customers) into the network (cf. service station); these packets are then transmitted (cf. served) over a network link if the link is free, or stored temporarily in a buffer memory (cf. queue) if the link is busy transmitting another packet.

Given such a queueing model, the question is how to evaluate performance measures of interest, such as the buffer overflow (or packet loss) probability. Ideally, one would like to do this using only *analytical* means, such as probability theory and calculus, in order to arrive at closed-form expressions of the performance measure in terms of the parameters of the model. Indeed, for simple models this is often possible; see [Kle75a] for many examples. Such models typically contain only one queue, and/or the distribution of the interarrival and service times have nice properties (e.g., memoryless distribution), and/or the performance measure is simple (e.g., the average waiting time). In slightly more complicated models, an explicit closed form expression may not be obtainable, but results may still be found by a *numerical evaluation* (e.g., a recursion, a matrix equation, or an inverse Laplace transform). Furthermore, in cases where an exact solution is not possible, *approximations* may be used to simplify the calculations; for example, limit behaviour for large buffer size may be calculated.

An important approximation technique is *large-deviations theory*, which deals with the behaviour of a system when it deviates far from its average behaviour. An overview of large-deviations theory and its applications to telecommunication systems is provided in [Wei95].

If analytical or numerical calculation is not possible and appropriate approximations are not available, *simulation* can often be useful. Stochastic simulation is the use of a computer program to sample (pseudo-random) values for the random variables in the model, thus constructing a "sample path": a sequence of events that could happen in the model. By repeating this, many different and independent sample paths are obtained. Then an estimate of the quantity of interest can be obtained by taking the average of its value over all of the simulated sample paths. Obviously, due to the randomness, this is only an approximation of the quantity's true value. Therefore, together with the estimate one also calculates the variance of the estimator: a measure for how accurate the estimate is likely to be (for more precise definitions, see, e.g., [KL91]); the more sample paths are simulated, the smaller the variance becomes (unless the variance does not exist (is infinite)). The big advantage of simulation is its generality: in contrast to most analytical and numerical methods, it poses no restrictions on the probability distributions involved; also, given enough computer capacity, very complicated models can be simulated and estimates can be obtained at any desired accuracy.

Simulation also has some important problems, though. First of all, it can be very time-consuming: in order to estimate the probability of an event of interest accurately, one needs to collect many (almost) independent observations of it in the simulation run. Furthermore, one such a simulation run in principle only yields an estimate for one set of values of the model's parameters (buffer size, arrival rate, etc.). In practice, one frequently needs estimates at many parameter values, for example to choose the buffer size such that the overflow probability is sufficiently low; in such cases, the simulation needs to be repeated for many values of the parameter. (It should be noted however, that using modern techniques [RM98], one simulation run can be sufficient to obtain estimates for a range of values of some of the model's parameters.)

One class of problems for which (standard) simulation is rather unsuitable, is those involving the estimation of probabilities of rare events, i.e., events which have a very low probability of occurrence (e.g., 10^{-6} or less). Such events are of much interest in queueing models of telecommunications systems, since these are typically designed to have very low packet loss probabilities to guarantee a good quality of service. These low probabilities imply that the system can be simulated for a long time without the event occurring even once. As noted above, an event needs to be observed many times during a simulation run for the estimate of its probability to be accurate; therefore, the estimation of rare-event probabilities requires impractically long simulation runs if no specialized techniques are used, such as those discussed in the next section.

1.3 Rare event simulation

Two classes of methods for rare event simulation are known: those based on *importance sampling*, and those based on *splitting*. Both of these methods involve modifying the simulation such that the rare event of interest occurs more frequent than it would do otherwise, and then mathematically compensating for the influence of these modifications to obtain the true probability. The methods differ in the type of modification. In importance sampling, the probability distributions of the model are modified to make the occurrence of the target event more frequent; in splitting methods, sample paths that reach an intermediate levelset (between the starting state and the rare target levelset) are split into several separate paths, thus also causing frequent observations of the target event. Both methods are described in some more detail below.

1.3.1 Importance sampling simulation

As noted above, the basis of importance sampling is modifying the probability distributions of the model. Formally, this is called a *change of measure*[1]. The word *tilting* is also used often, either to refer to a specific form of change of measure, or as a short synonym for change of measure; we will use it in the latter sense in this thesis.

Changing the probability distributions of the model implies that with any sample path, *two* probabilities can be associated: one using the original probability distributions, and one using the alternative probability distributions. The alternative distribution can (and should) be chosen such that in a simulation based on that distribution the target event is not rare, and thus observed frequently. To ensure a correct estimate of the target event's probability, the simulation program needs to keep track of the *likelihood ratio*: the ratio between a sample path's probability using the original distribution and its probability using the alternative distribution. Every time the target event is observed in the modified system, the likelihood ratio can be used to give a correct contribution to the event's probability estimate. For more details see, e.g., [Hei95] and [NSH00].

A crucial problem in importance sampling simulation is the proper choice of the simulation distributions. If chosen incorrectly, the resulting estimator may have a greater variance that the one from standard simulation; the variance can even become infinite. In that case, the estimator may also *appear* to be biased

[1] this terminology is based on using measure theory for the description of probabilities.

(i.e., its expectation may appear to be different from the true value) in a finite simulation run, although it theoretically still is unbiased. No general method exists to choose the optimal simulation distribution. It must be chosen such that no sample path which actually reaches the event and has a non-zero probability under the original distribution, gets a zero probability; otherwise, the estimator will be biased. A practical guideline is that the paths that are made more likely by the new distribution must not just be any paths to the event of interest, but must be those paths which form the most typical way to reach the event in the original system. In fact, it is known that if one chooses the new distributions such that they are identical to the original distributions *conditioned* on the occurrence of the event of interest, one gets an importance sampling estimator with zero variance. However, this knowledge cannot be applied in practice, since if one could calculate these optimal distributions, one would already know the probability of interest exactly. In practice, heuristics are often used: one tries to find (e.g., using large-deviations theory) what the most likely paths to the event of interest are in the original system, and then tries to modify the distributions such that they favor this (and similar) paths.

Because of the difficulty of properly choosing the simulation distributions, a substantial part of the literature on importance sampling consists of analytical calculations of the performance of a given simulation distribution for a given problem. Often, the asymptotic behaviour is discussed; i.e., how the variance of the importance sampling estimator for a given number of sample paths (observations) changes when the event is made rarer, e.g., by increasing the buffer size in the case of a buffer overflow model. A desired property is *asymptotic efficiency*, meaning that the relative error (defined as the estimate's standard deviation divided by the estimate itself) increases at most polynomially in some "rarity" parameter (e.g., the buffer size), while the probability of the event decreases exponentially.

As an alternative to the use of heuristics and/or formal mathematical proofs to choose the change of measure, a number of adaptive methods have been developed recently. Such methods use a series of simulation runs to search for the optimal change of measure and converge to it. Some examples of this are [DT93b], [DT93a], [AQDT95], [RM98], and [Lie99]. The obvious advantage of such methods is their general applicability: they can be used even if not enough insight into (or analysis of) the model is available to decide what the typical paths to the rare event are.

1.3.2 Splitting methods

In splitting simulation methods, which are perhaps best known under the name RESTART (REpetitive Simulation Trials After Reaching Thresholds), every sample path that reaches a given intermediate levelset is split into several sample paths, each of which gets a chance to reach the next intermediate levelset. By properly choosing the intermediate levels, the complete sample path to the rare overflow levelset can thus be broken down into many non-rare events, namely reaching the next level after (re)starting from the previous level. Thus, the estimation of the rare event probability decomposes into several estimations of probabilities that can easily be performed by standard simulation, and a calculation (multiplication) to combine them. However, computing the variance of the resulting estimator is not always straightforward and depends on details of the splitting method chosen; see, e.g., [Gar00].

Properly applying the splitting method involves deciding how many restart levels to use, where to put them, how many restarts to perform at every level, etc. In particular, the proper placement of the levels can be problematic in a multi-dimensional model (e.g., a queueing model with several nodes); choosing them wrong may lead to inefficient simulation.

The splitting method will not be considered further in this thesis; for more details, the reader is referred to [VAVA91], [VAVA99], [GF98], [GHSZ98], [Gar00], and references therein.

1.4 Problems studied in this thesis

Two main problems are studied in this thesis, although some other related problems also get some attention. These main problems are: consecutive cell loss, and overflows in networks of queues. They are introduced in the following.

1.4.1 Consecutive cell loss

Most research into loss models for ATM (and other queueing) systems has concentrated on the probability of the loss of an individual cell (packet, customer, etc.). Evidently, this is an important characteristic of a system, but it does not tell the whole story: the pattern of loss can also be important for the impact on the QoS. Consider for example non-realtime traffic, such as file transfer. Typically, such traffic is generated as IP (Internet Protocol) packets, which are split (because of their size) over several ATM cells. If at least one of the cells of the packet gets lost, the entire packet is retransmitted by a higher-layer protocol, such as TCP (Transmission Control Protocol). For this retransmission process it does not matter whether one or multiple cells are lost, as long as they all belong to the same packet. Compare this with the case of real-time traffic. Cell loss in such traffic typically leads to interpolations or error-recovery using redundancies at the receiving end; if several consecutive or close cells are lost, the interpolation or recovery will be less accurate or impossible, whereas loss of a single cell would hardly give a noticeable QoS degradation.

Not much literature seems to exist on cell loss patterns, and consecutive cell loss in particular. In [NH96], the consecutive cell loss probability is calculated analytically for $M/M/1$ queues, and an efficient (but heuristic) importance sampling simulation method for the estimation of this probability in $GI/GI/1$ queues is described. In [LA96], policies to optimize the consecutive-cell loss performance of a leaky-bucket admission system are studied. In [KS97], the consecutive cell loss probability in a queueing model of an ATM switch with bursty arrivals is studied, using stochastic activity networks. Furthermore, [Bon91] considers the loss (not necessarily consecutive) of a large fraction of a group of consecutive arrivals. The models in the latter two papers contain multiple sources, and the losses of cells from a given ("tagged") source are considered. Finally, [RMV96] gives an approximation for the consecutive cell loss probability in a model with bursty sources.

In the present thesis, the estimation of consecutive (cell) loss probabilities is considered. Using analytical methods, the consecutive cell loss probability is calculated for several simple queueing systems ($M/G/1$ and $G/M/m$), as well as the per-stream consecutive cell loss probability in a multi-source $M/M/1$ queue. Furthermore, an importance sampling simulation method is developed for estimating the consecutive cell loss probabilities in $M/G/1$ queues, that is provably asymptotically (for large numbers of consecutive cells) efficient. In the course of this research we obtain a number of other interesting results (solutions of subproblems) that can also have applications in other contexts.

Obviously, consecutive cell loss is just one of many interesting loss patterns; e.g., losing 6 out of 8 consecutive cells may in many cases have the same impact on the QoS as losing 6 completely consecutive cells. Because of time limitations and the complexities already encountered while studying just consecutive loss, these more general (and useful) problems are not studied in this thesis.

1.4.2 Overflows in networks of queues

The other main problem considered in this thesis is the estimation of overflow probabilities in networks of queues, in particular using importance sampling simulation. Models of practical packet-switching communication systems typically contain more than one queue (with arbitrary routing), so the estimation of overflow (and thus loss) probabilities of such networks has much practical relevance.

In the literature, importance sampling estimation of overflow probabilities in queueing networks has received much attention during the last decade. One of the first publications is [PW89], in which a heuristically motivated change of measure is proposed for the importance sampling estimation of the overflow probability of both a single queue, and of the total population in a network of queues. For tandem[2] Jackson[3] networks, this change of measure boils down to exchanging the arrival rate with the lowest (bottleneck) service rate; for other networks, a numerical minimization is needed. Experimentally, this change of measure is found to work well for single queues, but not always for networks containing two or more queues. In [Sad91], the asymptotic efficiency of this method for a single $GI/GI/1$ queue is proved. The complexity of determining the change of measure is reduced significantly by [FLA91], where an analytical alternative to the numerical minimization is demonstrated to find the change of measure for Jackson and some other networks. The experimental observation that this heuristic change of measure does not always work well for models with more than one queue is explained by [GK95], where the working of this heuristic for tandem Jackson networks is studied analytically, resulting in the determination of regions in the parameter space (arrival and service rates) in which the resulting simulation is or is not asymptotically efficient.

The above papers all consider a change of measure that is “static”: it does not depend on the state. In other words: the interarrival and service time distributions of the model are simply replaced by other distributions, but this replacement stays the same during the entire simulation. A different approach is used in [KN99] for estimating the overflow probability of the second buffer in a two-node tandem Jackson network: the change of measure used there makes the arrival and service rates depend on the content of the first buffer, i.e., *state-dependent*. Two other examples of this are [Hee98b] and [MR00].

Furthermore, in all of the above papers, the change of measure is determined (often heuristically) on the basis of some mathematical calculation. Adaptive methods, as mentioned in Section 1.3.1, have not been applied much to queueing models; notable exceptions are [DT93a] and [RM98], where they have been used for finding a state-independent change of measure for some queueing models.

In the present thesis, we focus on adaptive importance sampling methods for the estimation of network overflow probabilities, using both state-independent and state-dependent tiltings. Several such methods are developed and applied to many queueing network examples, with good results in most cases. Particularly noteworthy is the fact that the state-dependent tilting yields an efficient simulation in the tandem networks for which the heuristic (state-independent) tilting does not work well according to [GK95]. In order to help verify the correctness of the simulations, also a simple and efficient numerical method is developed for estimating overflow probabilities in small Jackson networks.

[2] i.e., the queues are arranged ”in series”, so a customer subsequently goes through all queues.

[3] i.e., the interarrival and service times are exponentially distributed.

1.5 Outline of this thesis

As noted above, in the course of the work a number of interesting sub-problems were identified and solved, the results of which can be useful in other contexts. Therefore, the author has decided to structure the thesis according to the methodology used (and thus the type of result): part I (Chapters 2 through 4) about analytical methods and results; part II (Chapters 5 and 6) about "traditional" importance sampling with asymptotic efficiency proofs; and part III (Chapters 7 and 8) about adaptive importance sampling methods. A more detailed description of the content of the chapters is as follows:

In **Chapter 2** a simple procedure is described to calculate overflow probabilities in small networks of queues with exponentially distributed interarrival and service times. In principle, these can be calculated by standard Markov chain theory, but we develop a procedure to significantly reduce the size of the matrices involved by exploiting some properties of the structure of the Markov chain for these queueing problems. The method is used to verify the correctness of simulation results obtained in Chapters 7 and 8.

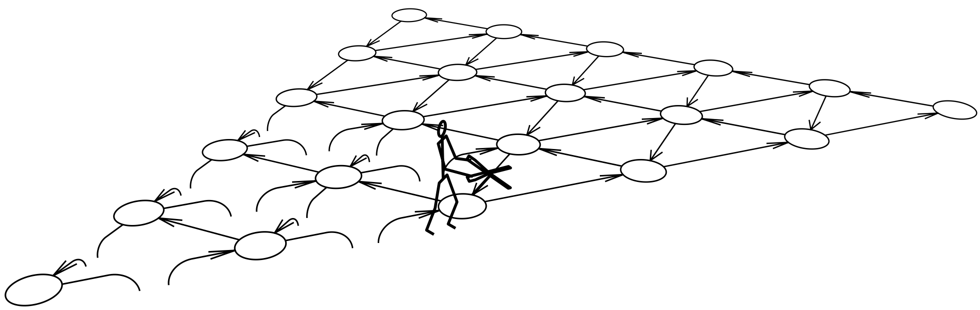
Chapter 3 describes the analytic calculation of consecutive cell loss probabilities and frequencies for some simple queueing systems. These are the $M/G/1$ and $G/M/m$ queues, and an $M/M/1$ queue where the losses of only one out of many input streams are considered. The obtained results are either explicit closed form expressions or can be easily evaluated numerically.

Chapter 4 demonstrates the calculation of the remaining service time distribution when the content of an $M/G/1$ queue reaches some high level (e.g., full buffer). Such remaining service times play a role in the consecutive cell loss problem: if several cells are lost consecutively, they all must arrive within the duration of a single full-buffer period, which is equal to this remaining service time. This chapter provides asymptotic results, valid in the limit of an infinitely high level; however, we show numerically that these results are also a good approximation at relatively low levels.

In both **Chapters 5 and 6**, asymptotically efficient importance sampling procedures are described for the estimation of certain probabilities involving sums of independently and identically distributed random variables. These arise in various problems, including the consecutive cell loss probability estimation. The chapters differ in the type of change of measure used. The change of measure used in Chapter 5 is less involved than the one in Chapter 6, but as a consequence it yields efficient simulation for a smaller class of problems. In Chapter 5, the consecutive cell loss problem is considered as an application example, while in Chapter 6 some other examples are presented.

Chapter 7 introduces the adaptive importance sampling method. In this chapter, only state-independent changes of measure are considered; i.e., the simulation distributions do not depend on the state of the system. Several variants of the method are proposed: a version which directly minimizes the variance, a computationally more efficient version which works indirectly by minimizing a cross-entropy function, and a version specific for problems that can be modelled by a discrete-time Markov chain. The methods are experimentally shown to be quite effective at finding a good change of measure for the simulation of several queueing network models. However, a few counterexamples are also demonstrated: cases in which no good state-independent change of measure can be found.

In **Chapter 8**, the adaptive importance sampling method based on cross-entropy is again applied to discrete-time Markov chains, but now the change of measure is allowed to depend on the state of the system. Although this in principle is a straight-forward extension, it poses several practical problems if the state space is large, as is typically the case in queueing network models. Some solutions to these problems are described in detail, followed by several experiments and a mathematical explanation for some of the phenomena observed. It is demonstrated experimentally that the method is asymptotically efficient in cases in which the state-independent method of Chapter 7 fails.



Chapter 2

Overflow probabilities in simple Jackson networks

This chapter describes a method for the calculation of (transient) overflow probabilities in simple Jackson queueing networks. Such networks can be used to model a packet switch in a telecommunications network (on the level of packets or calls), a multi-tasking computer system, a manufacturing system, etc. The main purpose of developing the method in this thesis is to provide reference solutions for small networks, to be used for validating the simulation algorithms developed in Chapters 7 and 8.

Although in this thesis the method is only applied to queueing network models, it can in fact handle a much larger set of discrete-time Markov chain models. It calculates the probability that one state of a set of "overflow" states is reached, starting from some initial state, and before reaching a set of absorbing states (e.g., empty-buffer states in the case of a queueing system). Precise definitions of this are given in the next section, together with a set of requirements the model must satisfy for the method to be applicable.

In principle, such probabilities are not hard to calculate using standard Markov chain theory. However, a straight-forward calculation typically involves inverting a matrix with as many rows and columns as there are states. The method described in the present chapter reduces the calulation to a large number of inversions of smaller matrices; thus, with the same computer capacity a larger state space can be handled (note that in practice, these calculations are only feasible numerically). For large enough models (e.g., models of networks with many queues) the size of the matrices can still become prohibitively large.

The material in this chapter is an extension of the method used in Appendix A.2 of [GK99]. There are also some similarities with matrix-geometric methods (see [Neu81]), but these are typically used for calculating steady-state probabilities, whereas the method discussed here calculates overflow probabilities of the type described above.

Section 2.1 describes the method in detail. Section 2.2 illustrates the method by applying it to some simple queueing models. Section 2.3 contains a few concluding remarks about the applicability of the method. No numerical results are provided: those can be found in Sections 7.4 and 8.3, where the method is used for the verification of simulation results.

2.1 The method

We begin by precisely specifying to what class of problems the method is applicable; it will become clear that the calculation of overflow probabilities in Jackson networks fits this class quite naturally. First of all, the method only applies to problems involving a discrete-time Markov chain (DTMC), in which the probability of interest is the probability that starting from some state (possibly only specified stochastically, i.e., a probability distribution over all states) a state belonging to a set of "overflow" states is reached before a state belonging to the set of absorbing states is reached. Furthermore, the state space of the DTMC is partitioned into (non-overlapping) level sets, where with every level set, an integer called the "level" is associated. This partitioning and the assignment of the levels must be done such that the following requirements are satisfied:

- Level 0 is the lowest level, and the corresponding level set is the set of the absorbing states.
- Level sets k and higher together form the set of the "overflow" states.
- At every transition of the Markov chain, the level may increment by at most 1. I.e., from a state at level i, only transitions to states at levels $0 \dots i+1$ can have a non-zero probability.
- All possible (i.e., having a non-zero initial probability) initial states must be in one level set, whose level we denote by n_0.

In fact, a trivial partitioning[1] into three level sets is always possible that meets all requirements; however, the method only simplifies the calculations significantly (by reducing the size of the matrices involved) if the number of levels is

[1] The trivial level sets are as follows: level 0 contains all absorbing states, level 2 contains all overflow states, and level 1 contains all other states (including the starting state). As can be verified easily, this partitioning satisfies all requirements, with $n_0 = 1$ and $k = 2$. Furthermore, it is also clear that this partitioning can be applied to any DTMC for which a probability of reaching some overflow state(s) before reaching some absorbing state(s) is sought.

large and (consequently) the number of states per level is small, so this trivial partitioning is not useful.

In queueing problems, it is usually quite easy to define (non-trivial) level sets such that the above requirements are all satisfied; typically, the level corresponds quite naturally to some observable quantity, like the total network population. We will see some examples of this in Section 2.2.

Coordinate transformation

For reasons that will become clear later, we need a two-dimensional coordinate system for the state space of the Markov chain. One of these coordinates must be the "level" defined above. The other coordinate, henceforth referred to as the *auxiliary* coordinate, must provide enough information to uniquely identify the state within a level set. Since the auxiliary coordinate is used as a row or column index in matrices, it needs to be a single integer. We will see some practical examples of suitably defining the auxiliary coordinate in Section 2.2.

In the sequel, we will refer to the state at level n with auxiliary coordinate i as simply "state i at level n".

One-step transition matrices $A_m(n)$

The one-step transition matrices $A_m(n)$ contain the one-step transitions probabilities of the DTMC. The transitions are sorted according to the levels of the states involved: the i,j element of $A_m(n)$ is the transition probability from state i at level n to state j at level $n+m$. Because no transition increases the level by more than 1, all $A_m(n)$ are 0 for $m > 1$.

Up-crossing matrices $Q(n)$

The i,j element of the up-crossing matrix $Q(n)$ is defined as the probability that, starting from state i at level n, level $n+1$ will be reached before level 0, and the entry state at level $n+1$ will be the one with auxiliary coordinate j.

Before considering the general form of $Q(n)$, first consider the slightly simpler case in which in a single transition of the Markov chain, the level cannot change by more than 1 (actually, this is true for many practical problems, including the ones considered in Section 2.2). By inspection, one finds:

$$Q(n) = A_{+1}(n) + A_0(n)Q(n) + A_{-1}(n)Q(n-1)Q(n).$$

Solving this for $Q(n)$ yields

$$Q(n) = \Big(I - A_0(n) - A_{-1}(n)Q(n-1)\Big)^{-1} A_{+1}(n), \tag{2.1}$$

where I is the identity matrix.

In the general case, we have

$$Q(n) = A_{+1}(n) + A_0(n)Q(n) + \sum_{m=-(n-1)}^{-1} A_m(n)Q(n+m)Q(n+m+1)\dots Q(n),$$

with solution

$$Q(n) = \left(I - A_0(n) - \sum_{m=-(n-1)}^{-1} A_m(n)Q(n+m)Q(n+m+1)\dots Q(n-1)\right)^{-1} A_{+1}(n). \tag{2.2}$$

Note that the sum over m extends down to $m = -(n-1)$; there is no need to extend the sum further, since by definition $Q(0) = 0$.

For $n = 1$, equation (2.2) directly gives $Q(1)$ in terms of $A_0(1)$ and $A_{+1}(1)$, since the sum over m becomes empty. Similarly, for $n = 2$, equation (2.2) expresses $Q(2)$ in terms of $Q(1)$ and $A_m(2)$. This can be continued, allowing subsequent calculation of $Q(n)$ for all $n \geq 1$.

Entrance probability vectors $\boldsymbol{\pi}(n)$

The entrance probability vector $\boldsymbol{\pi}(n)$ at level n is defined as follows: its ith element is the probability that the level n is reached before absorption (level 0) and that it is first entered at state i. Normalizing $\boldsymbol{\pi}(n)$ such that its elements sum up to 1 gives the entrance distribution of level n. On the other hand, summing the elements of $\boldsymbol{\pi}(n)$ gives the total probability of reaching level n before reaching 0.

As mentioned at the beginning of Section 2.1, the starting state is given by its level n_0, and a probability distribution over the states within that level; clearly, the latter is $\boldsymbol{\pi}(n_0)$. From this, the entrance probability vectors for higher levels can easily be calculated, as follows:

$$\begin{aligned}
\boldsymbol{\pi}(n_0+1) &= \boldsymbol{\pi}(n_0)Q(n_0) \\
\boldsymbol{\pi}(n_0+2) &= \boldsymbol{\pi}(n_0+1)Q(n_0+1) = \boldsymbol{\pi}(n_0)Q(n_0)Q(n_0+1) \\
&\vdots \\
\boldsymbol{\pi}(k) &= \boldsymbol{\pi}(n_0)Q(n_0)\dots Q(k-1).
\end{aligned}$$

Summing the components of $\boldsymbol{\pi}(k)$ completes the calculation of the overflow probability of level k.

2.2 A few simple queueing examples

2.2.1 The $M/M/1$ queue

The first example concerns the most trivial Jackson network possible: a single $M/M/1$ queue. The arrival rate is λ and the service rate is μ. We are interested in the probability that the buffer content reaches a high level k during one busy period (i.e., the time interval between two successive periods in which the buffer is empty).

An obvious choice for the "level" is the number of customers in the queue. Clearly, this choice satisfies all requirements: it is never incremented by more than 1, it is $\geq k$ in the overflow states, and it becomes zero at the end of the busy period (i.e., in the absorbing state). The initial level n_0 is 1: the level immediately after the first arrival to the empty queue.

Since the level as defined above completely determines the state of the queue, there is no need for the "auxiliary" coordinate. As a consequence, each of the A and Q matrices reduces to a scalar, making the calculations simple and analytically feasible (as opposed to just numerically).

Thus, we find the following values for the one-step transition "matrices", for all $n > 0$:

$$A_{+1}(n) = \frac{\lambda}{\lambda+\mu} = \frac{\rho}{\rho+1}, \qquad A_0(n) = 0, \qquad A_{-1}(n) = \frac{\mu}{\lambda+\mu} = \frac{1}{\rho+1},$$

where (as usual) $\rho = \lambda/\mu$. For the upcrossing "matrices" we find:

$$Q(n) = \begin{cases} \frac{\rho}{\rho+1} & \text{for } n = 1 \\ \frac{\rho}{\rho+1-Q(n-1)} & \text{for } n > 1. \end{cases}$$

As can be easily verified by substitution, the solution to the above recursion is

$$Q(n) = \frac{\rho^{-n}-1}{\rho^{-n-1}-1}.$$

Since there is no auxiliary coordinate, the entrance probability vectors $\boldsymbol{\pi}(n)$ also reduce to scalars, and $\boldsymbol{\pi}(1) = 1$. Thus, the overflow probability for level k is

$$\boldsymbol{\pi}(k) = \boldsymbol{\pi}(1)Q(1)Q(2)\ldots Q(k-1) = \frac{\rho^{-1}-1}{\rho^{-k}-1}.$$

This result is nothing new; see, e.g., [NH96].

2.2.2 Two queues in tandem

As the next example, consider the overflow probability of the total population in a (Jackson) network consisting of two queues in tandem, like the one depicted in Figure 2.1. Customers arrive at the first queue according to a Poisson-process with rate λ. Both servers have exponentially-distributed service times, with rates μ_1 and μ_2. The state of the system at any time is given by the two integers n_1 and n_2, which are the number of customers in the first and second queues, respectively.

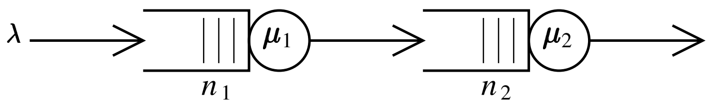


Figure 2.1: Two queues in tandem.

The probability of interest is the probability that the total population of the two queues reaches a given level L within a busy cycle. To be precise: we start with the system in the state $n_1 = 1, n_2 = 0$ (i.e., immediately after the first arrival of a busy cycle), and want to find the probability that $n_1 + n_2 = L$ is reached before $n_1 + n_2 = 0$.

Coordinates

A natural choice for the "level" is $n_1 + n_2$; it can easily be verified that this choice fulfills all requirements.

For the auxiliary coordinate, many choices are possible. Since the complete state of the system is given by the pair (n_1, n_2), and the sum $n_1 + n_2$ is already known as the level, any linear combination of n_1 and n_2 that is not a multiple of $n_1 + n_2$ would be suitable as the auxiliary coordinate. A practical choice is to simply choose n_1 as the auxiliary coordinate.

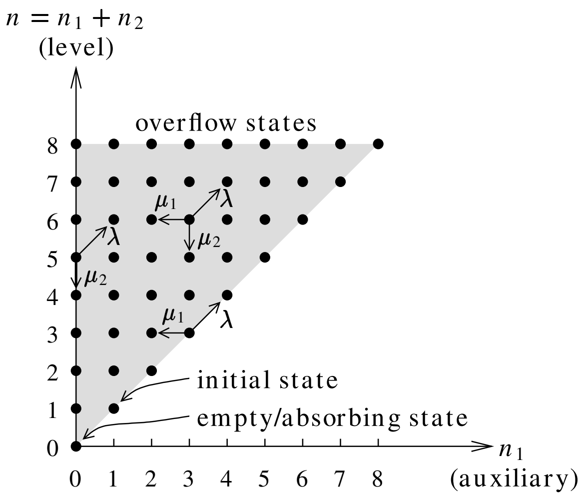


Figure 2.2: State space of two queues in tandem.

Figure 2.2 shows the set of possible states in the coordinates as just defined: the vertical axis shows the "level", the horizontal axis shows the auxiliary coordinate. Note that the latter (n_1) can never exceed the former ($n_1 + n_2$); this causes the set of possible states to have a triangular shape. The small arrows show the possible transitions in three typical states of the DTMC, along with their rates.

One-step transition matrices $A_m(n)$

It can easily be verified that the one-step transitions matrices for the two queues in tandem are as follows:

$$A_{+1}(n) = \left[\begin{array}{cccccccc} 0 & \frac{\lambda}{\lambda+\mu_2} & 0 & \ddots & 0 & 0 & 0 & \cdots \\ 0 & 0 & \frac{\lambda}{\lambda+\mu_1+\mu_2} & \ddots & 0 & 0 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots \\ 0 & 0 & 0 & \ddots & \frac{\lambda}{\lambda+\mu_1+\mu_2} & 0 & 0 & \cdots \\ 0 & 0 & 0 & \ddots & 0 & \frac{\lambda}{\lambda+\mu_1} & 0 & \cdots \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right],$$

$$\underbrace{\hphantom{\frac{\lambda}{\lambda+\mu_1+\mu_2}\quad\ddots\quad\frac{\lambda}{\lambda+\mu_1+\mu_2}}}_{n-1 \text{ columns}}$$

$$A_0(n) = \left[\begin{array}{cccccc} 0 & \ddots & 0 & 0 & 0 & \cdots \\ \frac{\mu_1}{\lambda+\mu_1+\mu_2} & \ddots & 0 & 0 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \cdots \\ 0 & \ddots & \frac{\mu_1}{\lambda+\mu_1+\mu_2} & 0 & 0 & \cdots \\ 0 & \ddots & 0 & \frac{\mu_1}{\lambda+\mu_1} & 0 & \cdots \\ 0 & \ddots & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right].$$

$$\underbrace{\hphantom{\frac{\mu_1}{\lambda+\mu_1+\mu_2}\quad\ddots\quad\frac{\mu_1}{\lambda+\mu_1+\mu_2}}}_{n-1 \text{ columns}}$$

and finally

$$A_{-1}(n) = \underbrace{\begin{bmatrix} \frac{\mu_2}{\lambda+\mu_2} & 0 & \ddots & 0 & 0 & \cdots \\ 0 & \frac{\mu_2}{\lambda+\mu_1+\mu_2} & \ddots & 0 & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \cdots \\ 0 & 0 & \ddots & \frac{\mu_2}{\lambda+\mu_1+\mu_2} & 0 & \cdots \\ 0 & 0 & \ddots & 0 & 0 & \cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots \end{bmatrix}}_{n-1 \text{ columns}}$$

Rest of the calculation

The next step would be the calculation of the up-crossing matrices $Q(n)$. Unfortunately, the up-crossing matrices do not have a nice and simple form, due to the matrix inversion involved. Therefore, this and the following steps are better left to a numerical calculation in a computer program.

Table 2.1 shows the listing of an implementation of these calculations in the Octave computer language (see [Oct]). Octave can directly manipulate vectors and matrices, which makes the program quite straightforward. Additional explanation is provided by the comments in the program: the text after # signs. Note that the program given here is more optimized for clarity than for efficiency.

The inner loop may need some more explanation. This loop calculates the one-step transition matrices $A_m(n)$ for level n, needed for the calculation of $Q(n)$. The loop constructs the matrices from scratch, in a way that can easily be generalised to more complicated systems. First, the three matrices `Adown`, `Asame` and `Aup` are initialised to zero. Then the loop considers all possible values of the auxiliary coordinate (called `i` in the program), in order to handle all states at level n. For each of these states, the total rate `t` of all possible transitions is calculated, after which the probabilities associated with those transitions are written into the appropriate elements of the matrices. For example, an arrival to the first queue increases both the level and the auxiliary coordinate `i` by one. So its probability must be written into the matrix `Aup` (that is $A_{+1}(n)$), at the position `i+1,i+2`. One might expect the position `i,i+1` here, but in the Octave language indices of matrices start from 1, whereas our auxiliary coordinate `i` can also be 0; therefore, all indices are displaced by 1 (alternatively, one could redefine the auxiliary coordinate to be $n_1 + 1$ instead of n_1).

```
#! /home/kam/ptdeboer/bin/octave

# Calculation of the overflow probability of the total network
# population in a two-node (M)/M/1 tandem queue.

# The parameters:
lambda = 0.04;     # arrival rate
mu1 = 0.48;        # service rate of first queue
mu2 = 0.48;        # service rate of second queue
L = 50;            # highest overflow level of interest
N = L+1;           # N^2 is the size of the matrices; N must be at least L+1

# Initialize some variables:
n = 0;             # the present value of the level
Qn = zeros(N,N);   # this is Q(n), i.e. Q(0) initially
pi = [0, 1, zeros(1,N-2)];  # this is pi(n+1), i.e. pi(1) initially

# Loop over all values of n from 1 up to L-1
while (n<L-1)

  # first increment n
  n = n+1;

  # now, calculate    A_{-1}(n), A_0(n) and A_{+1}(n),
  # which are called  Adown,     Asame  and Aup       in this program.
  # first initialize them to 0:
  Aup = Asame = Adown = zeros(N,N);
  # then loop over all values of the auxiliary coordinate (number of
  # customers in the first queue) and write all transition probabilities
  # into the matrices:
  for i=0:n
    # t is the total rate of all possible transitions from the present state:
    t = lambda;
    if (i>0) t = t+mu1; endif
    if (n-i>0) t = t+mu2; endif
    # write into the matrices:
    Aup(i+1,i+2) = Aup(i+1,i+2) + lambda/t;                  # arrival
    if (i>0) Asame(i+1,i) = Asame(i+1,i) + mu1/t; endif       # service at 1st
    if (n-i>0) Adown(i+1,i+1) = Adown(i+1,i+1) + mu2/t; endif # service at 2nd
  endfor

  # next, apply (2.1) to calculate Qn (note that up to here, the matrix Qn
  # is actually Q(n-1), since we've already incremented n):
  Qn = inverse( eye(N) - Asame - Adown*Qn ) * Aup;

  # then calculate pi(n+1) from pi(n):
  pi = pi*Qn;

   # finally, sum the components of pi(n+1) and print the result, which is the
   # probability of reaching level n+1:
  sum(pi)

endwhile
```

Table 2.1: Octave program for calculating the overflow probability of two queues in tandem.

2.2.3 More complicated networks

In both examples considered above, choosing the auxiliary coordinate was trivial: it was non-existant in the first example, and identical to n_1 in the second example. As soon as more than two queues are involved, things become a bit more complicated. Consider a network with three queues. The state of such a network is completely described by the triplet (n_1, n_2, n_3), where n_i is the number of customers in the ith queue. Assuming the event of interest is again the overflow of the total network population, a suitable definition for the level is $n_1 + n_2 + n_3$. At first glance, the auxiliary variable would need to be two-dimensional, e.g., the pair (n_1, n_2), in order to ensure that together with the level (a one-dimensional quantity) it could uniquely determine a point in the three-dimensional state space. However, the matrix formulation only allows a one-dimensional auxiliary coordinate. A solution would be to define the auxiliary variable as $n_1 + (L+1) \cdot n_2$, where L is the maximum possible value of n_1; clearly, this assigns a unique integer to each possible pair (n_1, n_2). Similarly, for four queues one could use $n_1 + (L_1 + 1) \cdot n_2 + (L_1 + 1)(L_2 + 1) \cdot n_3$, where L_i is the maximum possible value of n_i, etc.

In some problems, the choice of the level may also be non-trivial. Consider a network with two queues, where one is interested in the probability that the content of the *second* queue reaches some level K before *both* queues become empty. In this case, one cannot simply use the number of customers in the second queue as the "level": although this gives the correct set of overflow states (with $k = K$), it does not give the correct set of absorbing states. The only absorbing state is $(n_1 = 0, n_2 = 0)$, whereas level 0 in this definition would also include states in which the first queue in non-empty. A correct choice would be the following:

$$\text{level} = \begin{cases} n_2 + 1 & \text{if } n_1 + n_2 > 0 \\ 0 & \text{if } n_1 + n_2 = 0, \end{cases}$$

with overflow level $k = K + 1$, as can be easily verified.

2.3 Concluding remarks

In this short chapter, we have presented a simple method to efficiently calculate overflow probabilities in Jackson networks. This method works by first partitioning the state space into level sets satisfying certain conditions, and then performing some matrix calculations once for each level. Due to the state space partitioning, the sizes of the matrices involved are reduced compared to a direct calculation of these probabilities.

Below, a few remarks are given on the applicability of the method, also to other models than Jackson queueing networks. There are two issues: whether a given problem fits the structure required for the method to work, and whether the computations needed are practically feasible.

2.3.1 Types of system

As stated in the beginning of this chapter, the method discussed is only applicable to systems that can be modeled as a discrete-time Markov chain; this could of course be an embedded Markov chain, thus extending the method in principle to such queues as $M/G/1$ and $G/M/1$. However, being a DTMC is not enough: the problem must also be such that a nontrivial[2] "level" can be defined in accordance with the requirements put forward in Section 2.1. In particular, the requirement that the level never increases by more than 1 in one step of the Markov chain may be hard to meet in certain problems, such as those involving overflows in $M/G/1$ queues.

A nice property of the method is that it does not require any kind of regularity in the DTMC (apart from the possibity of defining non-trivial level sets): in principle, the number of states and the transition probabilities could depend on the level. As a consequence, systems in which some buffers have a limited capacity, or where a service rate depends on the number of customers (e.g., a system with multiple servers in parallel) pose no problem. For comparison, the matrix-geometric method needs a certain regularity in the system's structure to work (note that this is not really a fair comparison, since the matrix geometric method computes different quantities than ours).

2.3.2 Practical limitations

Consider a network with K queues, each with an overflow level of N. In principle, such a system has a K-dimensional state space (one integer per queue). Assuming that the event of interest is overflow of one particular queue, the number of customers in that queue would be chosen as the level. The auxiliary coordinate must describe the number of customers in the remaining $K-1$ coordinates: it must distinguish between a total of N^{K-1} different states. Consequently, each $A_m(n)$ and $Q(n)$ matrix has $N^{2(K-1)}$ elements.

The above means that except for systems with very few queues, the size of the matrices grows very rapidly with the overflow level. For example, in a system with 4 queues, the size of the matrix grows with the 6th power of the overflow level; a relatively low overflow level like 25 already requires matrices with about 244 million elements. Implemented straightforwardly, this requires lots of memory; furthermore, calculating the inverse of such a matrix is quite demanding in terms of CPU time.

[2] see footnote 1 on page 14.

However, the $A_m(n)$ matrices are typically sparse: most of their elements are zero. To some extent this also holds for the $Q(n)$ matrices. This sparseness could be exploited to save both memory and computational effort.

Finally, note what happens if the trivial partitioning described in the footnote on page 14 would be used. In that case, most of the calculation reduces to solving (2.1) for $n = 1$, which is $Q(1) = \left(I - A_0(1)\right)^{-1} A_{+1}(1)$, where $A_0(1)$ is a matrix with as many rows and columns as there are states at level 1; since almost all states are at level 1 if the trivial partitioning is used, this would be almost an $N^K \times N^K$ matrix, i.e., a total of about N^{2K} elements. So applying a non-trivial partitioning reduces the size of the matrices by a factor of N^2, or allows one to handle one more queue (i.e., increase K by 1) with the same matrix size.

Chapter 3

Consecutive loss in simple queues

In this chapter, one aspect of loss patterns will be studied: the probability of losing several consecutive cells, which is of interest for QoS guarantees in telecommunications systems. Not much literature seems to exist on this specific problem; an overview is given in Section 1.4.1.

The present chapter contributes analytical calculations of the consecutive cell loss probabilities and frequencies for three models. The first, in Section 3.1, is the $M/G/1/k$ queue (with a single source). These results will mostly be used in Chapter 5 to verify simulation results. Next, the $G/M/m/k$ queue is considered in Section 3.2. Finally, in Section 3.3 an $M/M/1$ queue with multiple sources (traffic streams) is considered, in which the consecutive cell loss probability for a given traffic stream is calculated.

Note that throughout this chapter, we talk about consecutive *cell* loss, in view of the application to models of ATM systems. However, the analysis is not ATM-specific, so one could also read "customer" or "packet" instead of "cell".

3.1 Consecutive loss in an $M/G/1/k$ queue

In this section, we calculate the probability that during one busy period (defined as the time between an arrival which finds the system empty and the first time the system becomes empty again after that), a group of n (or more) consecutive arrivals is lost at least once. Such a loss event is referred to as an n-CCL event.

Consider the embedded Markov chain for this queue, with embedded points immediately after every service completion. Since there is only room for k cells, and service completion implies departure of the cell in service, the number of cells at the embedded point cannot exceed $k-1$. We label the states according to the number of cells in the queue, i.e., from 0 through $k-1$. Figure 3.1 shows the state space and all possible transitions out of one typical state. The symbol p_i used in the figure denotes the probability of i arrivals during one service interval.

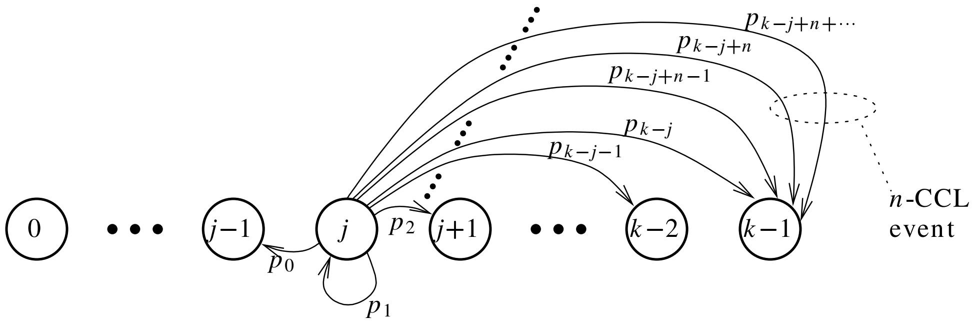


Figure 3.1: The embedded Markov chain, showing all states, but only transitions out of state j.

After a service completion (departure) there is at least one free place in the queue, so the first arrival after a service completion will surely be accepted. Therefore, in the above Markov chain the n-CCL event corresponds to a transition between two embedded points during which at least $n+m$ cells arrive, with m equal to the number of free places in the queue at the former embedded point. In other words: if the system is in state j at some embedded point, then the probability that the n-CCL event happens before the next embedded point is equal to the probability that at least $n+(k-j)$ cells arrive during the next service interval.

Define π_j as the probability that, starting from state j, the n-CCL event occurs before the queue becomes empty (i.e., before state 0 is reached). Then $\pi_0 = 0$, and the probability of interest is π_1. The following equality for π_j is easily found by inspection:

$$\pi_j = \sum_{i=0}^{k-j} p_i \pi_{j+i-1} + \sum_{i=k-j+1}^{k-j+n-1} p_i \pi_{k-1} + \sum_{i=k-j+n}^{\infty} p_i \cdot 1.$$

The three sums together cover all possible numbers i of arrivals between two embedded points. The first summation covers all cases in which so few cells arrive, that none are lost. The second summation covers the cases in which more cells arrive than can be accepted, so between 0 and n cells are lost; the n-CCL event does not occur (yet). The third summation covers the remaining cases, which are those in which the n-CCL event does occur before the next embedded point.

The above equality expresses π_j (left-hand side) in terms of $\pi_{j-1}, \pi_j, \ldots \pi_{k-1}$ (right-hand side). By solving for π_{j-1}, we can express it in terms of $\pi_j \ldots \pi_{k-1}$ as follows:

$$\pi_{j-1} = \frac{1}{p_0}\left(\pi_j - \sum_{i=1}^{k-j} p_i \pi_{j+i-1} - \sum_{i=k-j+1}^{k-j+n-1} p_i \pi_{k-1} - \sum_{i=k-j+n}^{\infty} p_i \cdot 1\right). \tag{3.1}$$

If we would know π_{k-1}, we could repeatedly apply the above equality, to first calculate π_{k-2} by substituting $j = k-1$, then π_{k-3}, and so on all the way to π_1, which is our probability of interest. Unfortunately, no boundary condition for π_{k-1} is available; we only have a boundary condition at the opposite end, namely $\pi_0 = 0$. Thus, we need to use the above equality to express all π_j, including π_0, in terms of π_{k-1}; then the boundary condition $\pi_0 = 0$ can be be applied to find the value of π_{k-1}, after which the value of the other π_j's follow. Note that (3.1) is such that all π_j will be of the form

$$\pi_j = \alpha_j + \beta_j \pi_{k-1}, \tag{3.2}$$

with α_j and β_j independent of π_{k-1}. Substituting this into (3.1) for π_{j-1} gives us the following recursions for α_j and β_j:

$$\alpha_{j-1} = \frac{1}{p_0}\left(\alpha_j - \sum_{i=1}^{k-j-1} p_i \alpha_{j+i-1} - \sum_{i=k-j+n}^{\infty} p_i \cdot 1\right)$$

and

$$\beta_{j-1} = \frac{1}{p_0}\left(\beta_j - \sum_{i=1}^{k-j-1} p_i \beta_{j+i-1} - \sum_{i=k-j}^{k-j+n-1} p_i\right).$$

To get the calculation of the α_j and β_j started, note that we have a boundary condition for them at $j = k-1$: from the trivial fact that $\pi_{k-1} = 0 + 1 \cdot \pi_{k-1}$, we find $\alpha_{k-1} = 0$ and $\beta_{k-1} = 1$. Starting from this, α_j and β_j can be calculated for all j down to 0. Then the original boundary condition $\pi_0 = 0$ can be applied, by substitution into (3.2), yielding $\pi_{k-1} = -\alpha_0/\beta_0$. Thus, we find

$$\pi_j = \alpha_j - \beta_j \frac{\alpha_0}{\beta_0},$$

which in principle completes the calculation. In general, the form of the explicit expressions for α_j, β_j and π_j, and thus the quantity of interest π_1, will be complicated, so this calculation is best done numerically. Example results of such a numerical evaluation are shown in Table 5.1 in Chapter 5, where they are used to validate simulation results.

Other quantities can also easily be calculated on the basis of the above. For example, consider the expected number of n-CCL events per busy cycle. To calculate this, note that immediately after any n-CCL event the system is in state $k-1$. Therefore, the probability that an n-CCL event is followed by another n-CCL

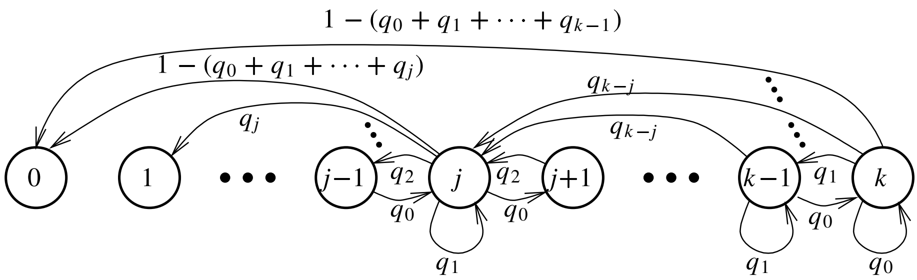


Figure 3.2: The embedded Markov chain for $G/M/1/k$, showing all states, but only transitions out of and into states j (valid for $0 \le j \le k-1$) and state k.

event before the end of the busy period is π_{k-1}. Thus the expected number of n-CCL events per busy cycle is

$$\pi_1 \cdot (1 + \pi_{k-1} + \pi_{k-1}^2 + \cdots) = \frac{\pi_1}{1-\pi_{k-1}}.$$

3.2 Consecutive loss in a $G/M/m/k$ queue

In a $G/M/m/k$ queue, the service time distribution is exponentially distributed. Because of the memoryless property of the exponential distribution, the duration of the full-buffer period (which equals the remaining service time at the moment the full-buffer state is reached) always has this same exponential distribution. Therefore, the probability that an n-CCL event happens in a given full-buffer period is independent of how the full-buffer state was reached. This observation simplifies the calculations below.

Consider the calculation of the probability that a given arrival happens to be the first arrival of an n-CCL event (this can also be interpreted as the frequency of the n-CCL event, see Section 3.3.4). A given arrival is the first arrival of an n-CCL event if and only if the following conditions are met:

(a) The previous arrival has found $k-1$ cells in the queue.

(b) No service completion occurs until after at least $n-1$ more cells have arrived (and thus been lost).

These are independent events, so we can just calculate their probabilities separately and multiply them.

(a) The probability of (a) is just the steady-state probability of state $k-1$ in the embedded Markov chain at arrival instants. This Markov chain is illustrated in Figure 3.2 for the case $m = 1$; extending this to $m > 1$ involves a slightly more complicated set of transition probabilities. The states are labelled according to the number of cells an arriving customer finds in the queue. The transition probabilities q_i are the probabilities of i service completions between two consecutive arrivals, given by

$$q_i = \int_0^\infty e^{-\mu x} \frac{(\mu x)^i}{i!} dF(x),$$

where $F(\cdot)$ is the distribution function of the interarrival times. By subsequently applying flow balance at the states of the Markov chain, one finds that the steady state probabilities π_j are given by

$$\pi_j = \frac{c_j}{c_0 + \cdots + c_k},$$

with

$$c_k = 1, \quad c_{k-1} = \frac{1 - q_0}{q_0},$$

and

$$c_{j-1} = \frac{1}{q_0} \left(c_j - \sum_{i=1}^{k-j} c_{i+j-1} q_i - c_k q_{k-j} \right)$$

for $j = k-1, \ldots, 1$. The probability of (a) then equals π_{k-1}.

(b) This condition means that the sum of n interarrival times is less than the remaining service time, which is distributed exponentially with rate $m\mu$ (assuming $k \geq m$; otherwise, set $m = k$). The probability is thus given by

$$\begin{aligned}
\mathbb{P}\left(\sum_{i=1}^{n} X_i < Z\right) &= \int_0^\infty \int_0^z dF_{\sum X_i}(s)\, m\mu e^{-m\mu z}\, dz = \int_0^\infty \int_s^\infty m\mu e^{-m\mu z} dz\, dF_{\sum X_i}(s) \\
&= \int_0^\infty e^{-m\mu s} dF_{\sum X_i}(s) = \tilde{F}_X^n(m\mu),
\end{aligned}$$

where X_i are the interarrival times, $F_{\sum X_i}$ is shorthand notation for the distribution function of $\sum_{i=1}^n X_i$, $\tilde{F}_X$ is the Laplace-Stieltjes transform of the interarrival time distribution, and Z is the duration of the full-buffer period.

The consecutive cell loss frequency now follows as the product of the probabilities of (a) and (b). Other probabilities of interest can be calculated from this.

Note: this section is a summary of material from [K+95], included here for completeness.

3.3 Per-stream consecutive loss in an $M/M/1$ queue

In this section, an $M/M/1/k$ queueing system with multiple (Poisson) input streams will be analysed. One of these input streams (sources) is the *foreground* stream, and we are interested in the consecutive (cell) loss as experienced by this foreground stream; therefore, the term n-CCL event now refers to losing n consecutive foreground arrivals. We will calculate the probability distribution of the number of n-CCL events between two foreground arrivals that find the queue empty, and (from this) the frequency of the n-CCL events.

3.3.1 Model and preliminaries

Consider a multiple input first-come first-served $M/M/1/k$ queue. The arrival rate for the "foreground" stream is λ_{f}. All other streams are combined into one "background" stream, whose arrival rate is λ_{b}. Thus, the total arrival rate $\lambda = \lambda_{\mathrm{f}} + \lambda_{\mathrm{b}}$, and the fraction of foreground arrivals in the aggregate arrival stream is given by $f = \lambda_{\mathrm{f}}/\lambda$. The service rate is μ, and the server utilization $\rho = \lambda/\mu$.

Start by modeling this queue by a simple Markov chain (actually, a birth-death process) where the states are labeled by the number of cells in the system; so we have states 0 through k, with state 0 corresponding to an empty system and state k being the full-buffer state.

Define:

- π_i = probability of reaching state k from state i, without hitting state 0. In [NH96], the analytical expression $\frac{\rho^{-i}-1}{\rho^{-k}-1}$ was derived for π_i.

- β_i = probability of reaching state 0 from state i, without hitting state k; obviously, $\beta_i = 1 - \pi_i$

- η = probability that after a full-buffer period (i.e., starting from state $k-1$) a new full-buffer period (state k) will be reached before any foreground arrivals. That is, no foreground cells are accepted between these full-buffer periods.

- ϕ_i = probability of reaching state k from state i, without hitting state 0 and without any foreground arrivals.

- ψ_i = probability of reaching state 0 from state i, without hitting state k and without any foreground arrivals.

- For convenience: $\tau = \lambda_{\mathrm{b}} + \lambda_{\mathrm{f}} + \mu$.

All of these quantities can be calculated relatively easily. Start by writing down the obvious recursive relation for ϕ_i:

$$\phi_i = \frac{\mu}{\tau}\phi_{i-1} + \frac{\lambda_{\mathrm{b}}}{\tau}\phi_{i+1} + \frac{\lambda_{\mathrm{f}}}{\tau}\cdot 0, \quad \text{for} \quad 1 \le i \le k-1. \tag{3.3}$$

Exactly the same recursion holds for ψ_i, the only difference between ϕ_i and ψ_i being the boundary conditions, which are

$$\phi_0 = 0, \quad \phi_k = 1 \quad \text{and} \quad \psi_0 = 1, \quad \psi_k = 0.$$

In order to solve the recursion, try substituting $\phi_i = z^i$ into (3.3), which yields the following condition for z: $\mu - \tau z + \lambda_b z^2 = 0$, with solutions

$$z_\pm = \frac{\tau \pm \sqrt{\tau^2 - 4\lambda_b \mu}}{2\lambda_b}. \tag{3.4}$$

Observe that $z_+ z_- = \frac{\mu}{\lambda_b}$. Taking the boundary conditions into account, one finds:

$$\phi_i = \frac{z_+^i - z_-^i}{z_+^k - z_-^k}$$

and

$$\psi_i = \frac{z_+^k z_-^i - z_-^k z_+^i}{z_+^k - z_-^k} = \left(\frac{\mu}{\lambda_b}\right)^i \phi_{k-i}.$$

Finally, we can express η in terms of the ϕ_i and ψ_i as follows:

$$\begin{aligned}
\eta &= \phi_{k-1} + \psi_{k-1} \frac{\lambda_b}{\lambda_b + \lambda_f} \sum_{i=0}^{\infty} \left(\psi_1 \frac{\lambda_b}{\lambda_b + \lambda_f}\right)^i \phi_1 \\
&= \phi_{k-1} + \psi_{k-1}\phi_1 \frac{\lambda_b}{\lambda_b + \lambda_f} \frac{1}{1 - \psi_1 \frac{\lambda_b}{\lambda_b + \lambda_f}} \\
&= \phi_{k-1} + \frac{\lambda_b \phi_1 \psi_{k-1}}{\lambda - \psi_1 \lambda_b}.
\end{aligned}$$

To see this, recall that η is the probability of reaching state k from state $k-1$ without any foreground arrivals. This can happen either without reaching state 0 (probability ϕ_{k-1}), or by first reaching state 0 (probability ψ_{k-1}), then having a background arrival to go to state 1, then any number i (with $i \geq 0$) of returns to state 0 each followed by a background arrival, and eventually reaching the full-buffer state again (probability ϕ_1).

3.3.2 The embedded Markov chain

To ease the further analysis of the problem, we define a new embedded Markov chain. Its embedded points are those times at which a foreground cell arrives and either finds the system full (and thus is rejected) or finds the system empty. The latter event is defined to be the starting point of a new regeneration cycle. Note that such a regeneration cycle can contain several busy periods of the queue, which is necessary because the n-CCL event of one stream can be spread over several busy periods. This embedded Markov chain has the following states (see Figure 3.3):

- 'R': a new regeneration cycle has been started.
- 'i': the last i foreground cells have been rejected, $1 \leq i \leq n$.

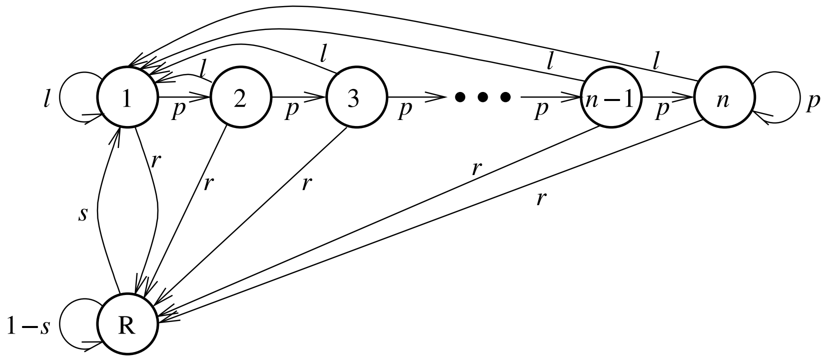


Figure 3.3: The embedded Markov chain.

Clearly, the occurrence of the loss of (at least) n consecutive foreground cells corresponds to a transition from state $n-1$ to state n. To find the distribution of the number of times this happens in a regeneration cycle, we first need to find the transition probabilities in this Markov chain.

The transition probability p

The probability p of going from state i to state $i+1$, for $i < n$, is the probability of losing the next foreground arrival, starting in the full-buffer state. By inspection, one easily finds that

$$p = \frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f} + \mu} \sum_{j=0}^{\infty} \left(\frac{\mu}{\lambda_\mathrm{f} + \mu} \cdot \eta \right)^j = \frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f} + \mu(1-\eta)},$$

where j is interpreted as the number of times a full-buffer period ends and a new full-buffer period is reached before the next foreground arrival. The resulting expression for p can also be justified intuitively, by interpreting $\mu(1-\eta)$ as an "effective" departure rate: only a fraction $(1-\eta)$ of all departures are followed by the acceptance of a foreground cell before full buffer is reached again.

The transition probabilities r and l

For r, similar reasoning as for p can be applied, although the result is a bit more complicated:

$$r = \frac{\mu}{\lambda_\mathrm{f} + \mu} \sum_{j=0}^{\infty} \left(\pi_{k-1} \cdot \frac{\mu}{\lambda_\mathrm{f} + \mu} \right)^j \beta_{k-1} \sum_{j=0}^{\infty} \left(\frac{\lambda_\mathrm{b}}{\lambda_\mathrm{b} + \lambda_\mathrm{f}} \cdot \beta_1 \right)^j \left(\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{b} + \lambda_\mathrm{f}} + \frac{\lambda_\mathrm{b}}{\lambda_\mathrm{b} + \lambda_\mathrm{f}} \pi_1 r \right).$$

This is explained by noting that starting from a full-buffer period, the regeneration is in general reached through the following steps:

1. the full-buffer period ends without any more foreground arrivals (probability $\mu/(\lambda_f + \mu)$);
2. the full-buffer state is reached (π_{k-1}) and left again without foreground arrivals ($\mu/(\lambda_f + \mu)$), a total of j times;
3. the buffer empties (β_{k-1});
4. j (in general different from the j in step 2) cycles happen, each consisting of a background arrival ($\lambda_b/(\lambda_b + \lambda_f)$) and subsequent emptying of the system without reaching full-buffer (β_1);
5. either a foreground arrival occurs ($\lambda_f/(\lambda_b + \lambda_f)$), thus providing the regeneration, or a background arrival ($\lambda_b/(\lambda_b + \lambda_f)$) is followed by a climb to full buffer (π_1), followed by a repetition of this entire process finally leading to regeneration (r).

The above expression for r can be rewritten (recall that $\beta_i = 1 - \pi_i$):

$$\begin{aligned} r &= \frac{1}{\frac{\lambda_f+\mu}{\mu} - \pi_{k-1}} \cdot \beta_{k-1} \cdot \frac{1}{1 - \beta_1 \frac{\lambda_b}{\lambda_f+\lambda_b}} \cdot \frac{1}{\lambda_f + \lambda_b}(\lambda_f + \lambda_b \pi_1 r) \\ &= \frac{1}{\frac{\lambda_f}{\mu\beta_{k-1}} + 1} \cdot \frac{1}{\lambda_f + \pi_1 \lambda_b} \cdot (\lambda_f + \lambda_b \pi_1 r). \end{aligned}$$

Solving this for r yields

$$r = \frac{\lambda_f}{\left(\frac{\lambda_f}{\mu\beta_{k-1}} + 1\right) \cdot (\lambda_f + \pi_1 \lambda_b) - \lambda_b \pi_1} = \frac{\mu\beta_{k-1}}{\lambda_f + \lambda_b \pi_1 + \mu\beta_{k-1}}.$$

The transition probability l can now obviously be calculated from

$$l = 1 - p - r.$$

The transition probability s

First, define θ as the probability of losing a foreground cell during one *busy* cycle. One finds:

$$\theta = \pi_1 \sum_{i=0}^{\infty} \left(\frac{\mu}{\lambda_f + \mu}\pi_{k-1}\right)^i \frac{\lambda_f}{\lambda_f + \mu} = \frac{\lambda_f \pi_1}{\lambda_f + \mu(1 - \pi_{k-1})} = \frac{\lambda_f \pi_1}{\lambda_f + \mu\beta_{k-1}}.$$

Then s, which (see Figure 3.3) is the probability of losing a foreground cell during one *regeneration* cycle, is given by:

$$s = \theta \cdot \sum_{i=0}^{\infty} \left((1 - \theta)\frac{\lambda_b}{\lambda_b + \lambda_f}\right)^i = \frac{\lambda_b + \lambda_f}{\frac{\lambda_f + \mu\beta_{k-1}}{\pi_1} + \lambda_b} = \frac{(\lambda_b + \lambda_f)\pi_1}{\lambda_f + \lambda_b \pi_1 + \mu\beta_{k-1}}.$$

In the following, we will also need the ratio of s and r; based on the above, this is simply

$$\frac{s}{r} = \rho\frac{\pi_1}{\beta_{k-1}} = \rho\,\frac{\frac{\rho^{-1}-1}{\rho^{-k}-1}}{1-\frac{\rho^{-(k-1)}-1}{\rho^{-k}-1}} = \rho^k.$$

This completes the calculation of the transition probabilities of the embedded Markov chain.

3.3.3 Number of n-CCL events in a regeneration cycle

In order to calculate the probability distribution of the number of n-CCL events in a regeneration cycle, it is convenient to first calculate the probabilities q_n, defined as the probability that starting from state 1, state n is reached before regeneration. These probabilities can be calculated as follows:

$$q_n = p^{n-1} + \left(1 + p + p^2 + \cdots + p^{n-2}\right) \cdot l\, q_n = p^{n-1} + \frac{1-p^{n-1}}{1-p} \cdot l\, q_n,$$

so

$$q_n = \frac{p^{n-1}(1-p)}{1-p-l(1-p^{n-1})} = \frac{p^{n-1}(1-p)}{r+lp^{n-1}}$$

Looking at the Markov chain, one sees that the probability of having at least j n-CCL events in one regeneration cycle is

$$\mathbb{P}(O_n \geq j) = s \cdot q_n \cdot \left(\frac{lq_n}{1-p}\right)^{j-1}.$$

From this, we calculate the expected number of n-CCL events per regeneration cycle:

$$\begin{aligned}\mathbb{E}(O_n) &= \sum_{j=1}^{\infty} s \cdot q_n \cdot \left(\frac{l}{1-p}q_n\right)^{j-1} = s \cdot q_n \cdot \frac{1}{1-\frac{lq_n}{1-p}} \\ &= \frac{s}{r}(1-p)p^{n-1} = \rho^k(1-p)p^{n-1}.\end{aligned}$$

3.3.4 Consecutive cell loss frequency

In accordance with [NH96], we define the consecutive cell loss frequency $\mathcal{F}_n$ as the reciprocal of the average number of foreground arrivals between two subsequent n-CCL events. Alternatively, this can be interpreted as the fraction of foreground arrivals that happen to be the first (or alternatively, the n-before-last) of a series of at least n consecutive foreground losses. From [NH96] we also have

$$\mathcal{F}_n = \frac{\mathbb{E}(O_n)}{\mathbb{E}(N)},$$

where O_n is the number of n-CCL events in a regeneration cycle, and N is the number of foreground arrivals during a regeneration cycle. Since $\mathbb{E}(O_n)$ has already been calculated above, only the calculation of $\mathbb{E}(N)$ remains. For an $M/M/1/k$ queue, it is known (e.g., Section 3.6 in [Kle75a]) that the probability that the server is found idle by an arriving cell (or an arriving foreground cell) is given by $p_{\text{idle}} = \frac{1-\rho}{1-\rho^{k+1}}$. Consequently, the expected number of foreground arrivals in a regeneration cycle (which is delimited by foreground arrivals finding the system empty) is

$$\mathbb{E}(N) = \frac{1}{p_{\text{idle}}} = \frac{1-\rho^{k+1}}{1-\rho}.$$

Thus, we find for the consecutive cell loss frequency

$$\mathcal{F}_n = \frac{\mathbb{E}(O_n)}{\mathbb{E}(N)} = \frac{1-\rho}{1-\rho^{k+1}}\rho^k(1-p)p^{n-1} = p_{\text{FB}}(1-p)p^{n-1}, \tag{3.5}$$

where p_{FB} is the steady-state full buffer probability in an $M/M/1/k$ queue, which is given by

$$p_{\text{FB}} = \frac{1-\rho}{1-\rho^{k+1}}\rho^k.$$

Actually, (3.5) has a very simple interpretation: it is the probability that a given (random) foreground cell finds the queue in the full-buffer state, that the next $n-1$ foreground arrivals do so too, and that the next foreground arrival does *not*, thus ensuring that the present foreground loss was the n-before-last of a series of at least n foreground losses; this probability is just $\mathcal{F}_n$, as noted at the beginning of this section.

3.3.5 Asymptotic results

Low foreground traffic intensity

If the foreground arrival rate λ_{f} is much smaller than the background arrival rate ($\lambda_{\text{f}} \ll \lambda_{\text{b}}$), one can assume that on average, many background arrivals and service completions will happen between two foreground arrivals. As a consequence, the foreground cells just see random and approximately *independent* "snapshots" of the system. Under this assumption, the probability that a foreground cell is lost can be approximated by the steady-state full-buffer probability of the queue, which is p_{FB} as discussed above.

As noted at the beginning of Section 3.3.4, $\mathcal{F}_n$ is equal to the probability that a random foreground cell is the first of a series of at least n consecutive foreground losses. Under the above approximate independence assumption, this is given by

$$\mathcal{F}_n \approx (1-p_{\text{FB}})p_{\text{FB}}^n\,.$$

The $(1-p_{\text{FB}})$ factor comes from the fact that this cell is the *first* one to be lost; i.e., the previous foreground cell found the system not in full-buffer state.

High values of k

In order to calculate this limit, we need to go back to section 3.3.1 and calculate the limit behaviour of all quantities introduced there. For convenience, we will use the approximate-equals sign $\approx$ to denote that the ratio of the left- and right-hand sides is 1 in the limit $k \to \infty$.

Starting from equation (3.4), we find the following bounds for z_+ and z_-, based on the assumptions that $\lambda_\mathrm{f} > 0$, $\lambda_\mathrm{b} > 0$ and (stability condition) $\mu > \lambda_\mathrm{f} + \lambda_\mathrm{b}$:

$$\begin{aligned} z_+ &= \frac{\lambda_\mathrm{f} + \lambda_\mathrm{b} + \mu + \sqrt{\lambda_\mathrm{f}^2 + \lambda_\mathrm{b}^2 + \mu^2 + 2\lambda_\mathrm{f}\lambda_\mathrm{b} + 2\lambda_\mathrm{f}\mu - 2\lambda_\mathrm{b}\mu}}{2\lambda_\mathrm{b}} \\ &> \frac{\lambda_\mathrm{b} + \mu + \sqrt{\mu^2 - 2\lambda_\mathrm{b}\mu + \lambda_\mathrm{b}^2}}{2\lambda_\mathrm{b}} = \frac{\mu}{\lambda_\mathrm{b}} > 1, \end{aligned}$$

and similarly

$$0 < z_- < 1.$$

Thus, high powers of z_- tend to 0, and high powers of z_+ tend to infinity. Using this, one can see that for large k:

$$\phi_1 \approx \frac{z_+ + z_-}{z_+^k}, \qquad \phi_{k-1} \approx \frac{1}{z_+},$$

$$\psi_1 \approx \frac{\mu}{\lambda_\mathrm{b} z_+}, \qquad \psi_{k-1} \approx 0,$$

$$\eta \approx \frac{1}{z_+}.$$

Then

$$p \approx \frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f} + \mu(1 - 1/z_+)}$$

and consequently

$$\mathcal{F}_n \approx \rho^k (1-\rho) \left(\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f} + \mu(1 - 1/z_+)} \right)^{n-1} \frac{\mu(1 - 1/z_+)}{\lambda_\mathrm{f} + \mu(1 - 1/z_+)}. \tag{3.6}$$

3.3.6 Numerical results

In this section, the expressions for $\mathcal{F}_n$ derived above are evaluated numerically for several values of the parameters, to illustrate the dependence of the n-CCL frequency on those parameters and to test the quality of the approximations given in Section 3.3.5.

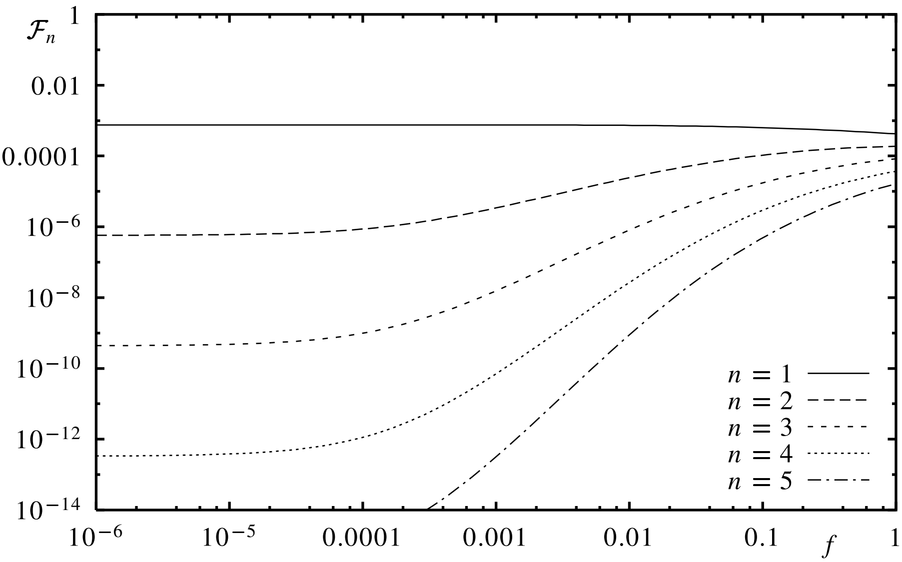


Figure 3.4: The n-CCL frequency $\mathcal{F}_n$ as a function of $f = \lambda_f/\lambda$; $\lambda = 0.8$, $\mu = 1$, $k = 25$.

n	$f = 1$	$f = 0.01$	$f = 0.0001$	$f = 10^{-6}$	limit
1	0.0004210	0.0007327	0.0007570	0.0007573	0.0007573
2	0.0001871	$2.431 \cdot 10^{-5}$	$8.648 \cdot 10^{-7}$	$5.816 \cdot 10^{-7}$	$5.739 \cdot 10^{-7}$
3	$8.317 \cdot 10^{-5}$	$8.063 \cdot 10^{-7}$	$9.880 \cdot 10^{-10}$	$4.466 \cdot 10^{-10}$	$4.350 \cdot 10^{-10}$
10	$2.849 \cdot 10^{-7}$	$3.565 \cdot 10^{-17}$	$2.509 \cdot 10^{-30}$	$7.039 \cdot 10^{-32}$	$6.246 \cdot 10^{-32}$

Table 3.1: Comparison of $\mathcal{F}_n$ for small $f = \lambda_f/\lambda$ and theoretical limit; $\lambda = 0.8$, $k = 25$.

Varying the fraction of foreground traffic

Figure 3.4 shows the n-CCL frequency as a function of the fraction f of foreground traffic in the input stream, for several values of n. Clearly, if the foreground traffic is very small in comparison to the background, the cell loss frequency converges to a constant. This agrees with our analysis in Section 3.3.5. Table 3.1 shows the numbers.

For $n \geq 2$, the n-CCL frequency clearly increases considerably with increasing f. This is intuitively reasonable: increasing f means that there will on average be fewer background arrivals between two consecutive foreground arrivals, causing a stronger correlation between the states in which consecutive foreground arrivals find the system; thus, it becomes more likely that after one foreground cell is lost, the next foreground cell will also be lost.

For $n = 1$, the n-CCL frequency is seen to decrease slightly with increasing f. This can best be considered an artifact of our definition of the n-CCL frequency. To see this, first note that the fraction of foreground cells that are lost is just the queue's steady-state full-buffer probability, which does not depend on f. Therefore, if every foreground loss would count as a 1-CCL event, the frequency $\mathcal{F}_1$ would be independent of f. However, our definition of the n-CCL event is such that if multiple, say n, cells are lost consecutively, these n foreground losses together only count as *one* 1-CCL event. Above, we already noted that at increasing f, such n-CCL events with $n > 1$ become more frequent; therefore, with increasing f, the 1-CCL frequency decreases.

Varying the buffer size

Figure 3.5 shows the n-CCL frequency as a function of the buffer size k, for several values of n. Also, the theoretical asymptotic curves from equation (3.6) have been plotted. Clearly, the asymptotic curves are approached reasonably fast for increasing k. However, additional experiments have shown that for larger ρ, the approximation is only good for larger k.

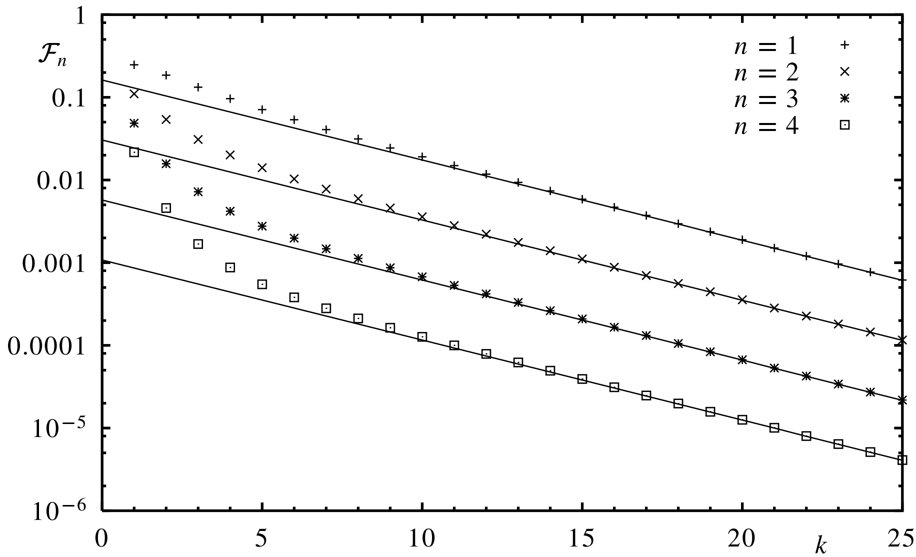


Figure 3.5: The n-CCL frequency as a function of k and the theoretical limit for large k (solid lines); $\lambda_b = 0.7$, $\lambda_f = 0.1$, $\mu = 1$.

Varying the number of consecutive cells lost

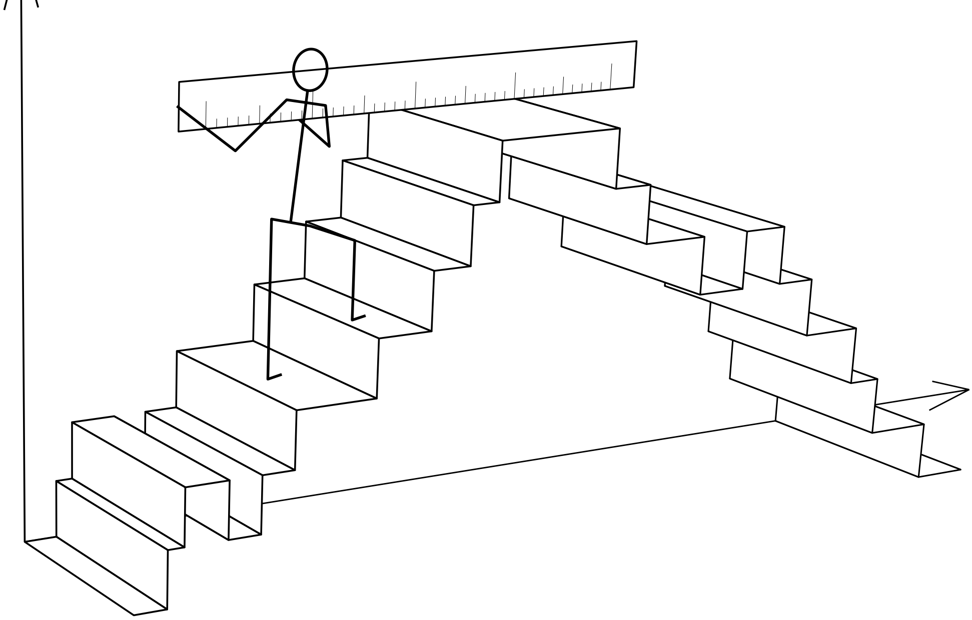
From equation (3.5), it is clear that $\mathcal{F}_n$ decays exponentially with n. This can also be seen in Figures 3.4 and 3.5, since curves for several values of n have been plotted.

3.4 Concluding remarks

In this chapter, some probabilities and distributions involving consecutive loss in simple queueing systems have been determined analytically. However, one may need to resort to numerical evaluation for practical applications.

In this chapter, we have first studied consecutive loss in $M/G/1/k$ and $G/M/m/k$ queues; these calculations turned out to be rather straightforward. These calculations rely on embedded Markov chains, so for extension of these results to more general queues, a different method would need to be developed.

Furthermore, we have demonstrated the calculation of the consecutive loss probability for one stream in an $M/M/1$ queue which is serving multiple independent streams of traffic. This analysis is quite complicated, even though it is a purely Markovian model. These complications are caused by the fact that in a multiple-stream model, the consecutive loss event is no longer confined to a single full-buffer period, or even to a single busy cycle. It may be possible to extend the present analysis to a $G/M/m$ queue with Poisson background traffic (i.e., only the foreground inter-arrival times would have a non-exponential distribution). For any further extension, the embedded Markov chain used in Section 3.3.1 is no longer well-defined, so a different approach would be needed; in such cases, simulation may be a more suitable technique.



Chapter 4

The remaining service time upon reaching a high level in $M/G/1$ queues

In this chapter, we study the distribution of the remaining service time upon reaching a high level (typically corresponding to full buffer) due to a customer arrival in an $M/G/1$ queueing system. This problem is motivated by research on efficient simulation of cell loss in such queueing systems (see Chapter 5) and could also be of interest in other contexts.

Consider an $M/G/1/B$ queue without service interruptions. Initially, assume that it is empty, i.e., there are neither customers waiting nor in service. After some time, the queue may become full, i.e., there are a total of B customers in it, one of which is being served. We are interested in the distribution of the remaining service time of the customer being served at the moment full buffer is reached. After the full-buffer state is left, the queue will sooner or later either become empty (marking the end of the busy cycle), or reach full buffer again during the same busy cycle; more full-buffer periods may follow in the same busy cycle. Because of the memoryless arrival process, the second and later full-buffer hits are stochastically equivalent, so we will refer to them as *subsequent hits* in this chapter. The first full-buffer hit in a busy cycle is in general different from subsequent full-buffer hits, and will be referred to as the *first hit*.

A huge amount of literature exists on the study of the single-server queue with all its variants; however, little is related to this problem. The closest we found was a discussion of the distribution of idle periods in a stable $GI/M/1$ queue in Chapter II.5.10 of [Coh82]. The stable $GI/M/1$ queue is the dual of the unstable $M/G/1$ queue, so those idle periods correspond to the remaining service times for 'subsequent' hits to full buffer in an unstable $M/G/1$ queue. Our analysis is more comprehensive, as it treats the stable as well as the unstable $M/G/1$ queue, and also the 'first' as well as 'subsequent' hits. In Chapter III.6.3 of [Coh82], there is a discussion of a related subject: the stationary joint distribution of the number of customers and the past service time in an $M/G/1$ queue. In [Asm81] the equilibrium distributions of the past and remaining service times upon arrival to a given level in an $M/G/1$ queue are calculated; equilibrium here implies that no distinction between first and subsequent hits is made: they are "mixed" according to the frequency with which they occur. Finally, in [Fak82] the expected value of the remaining service time upon arrival to a given level in $G/G/1$ queues is studied.

We start by introducing some notation in Section 4.1. Next, we derive some results for a hypothetical "doubly-unbounded" $M/G/1$ queue in Section 4.2. These results are used in Section 4.3 to find approximate results (accurate for large B) for the real bounded $M/G/1/B$ queue. Those results allow us to calculate the distributions of past and remaining service times in Section 4.4. However, this analysis does not hold for systems where the average service time equals the average inter-arrival time; to derive results for this case, we use a limit procedure in Section 4.5. As a by-product of the analysis in this chapter, we can also obtain an (asymptotically tight) approximation for the probability of reaching full buffer in a busy cycle, as demonstrated in Section 4.6. Section 4.7 illustrates the accuracy of our results by comparing them with results from exact numerical analysis and simulation. We present a summary of the results together with conclusions in Section 4.8. Note: the results in this chapter are only valid if a technical condition is satisfied (see (4.4)); which exclude cases where the service time distribution has a heavy tail.

4.1 Notation

Throughout this chapter, we will use some notational conventions which are introduced here. First, three generic random variables are defined:

- X is the (total) service time.
- Y is past service time upon hitting full buffer.
- Z is the remaining service time upon hitting full buffer.

Note that the distributions of Y and Z can be defective (in cases where there is a non-zero probability that full buffer is not reached in a given busy cycle); the defect will be represented by a probability mass at $+\infty$. We will also consider the distributions of Y and Z conditional on reaching full buffer, and denote these by $Y|\mathrm{fb}$ and $Z|\mathrm{fb}$, respectively; these are non-defective, of course.

For probability distributions the following notation is used, using the random variable W as an example:

- $f_W(\cdot)$ is the probability density function.
- $F_W(\cdot)$ is the distribution function.
- $\bar{F}_W(\cdot)$ is the complementary distribution function: $\bar{F}_W(w) = 1 - F_W(w)$.
- $\tilde{F}_W(\cdot)$ is the Laplace-Stieltjes transform of $F_W(\cdot)$: $\tilde{F}_W(s) = \int_0^\infty e^{-st} dF_W(t)$.

The arrival process is Poisson; its arrival rate is denoted by λ, and the system load is denoted by ρ, with $\rho = \lambda\mathbb{E}(X)$.

Finally, the symbol for approximate equality ($\approx$) in this chapter is understood to imply equality in the limit of infinite buffer size B; i.e., the limit for $B \to \infty$ (sometimes $i \to \infty$) of the quotient of the left-hand side and the right hand side is 1.

4.2 The doubly-unbounded $M/G/1$ queue

In this section, we study the "doubly-unbounded" $M/G/1$ queue. This hypothetical system is identical to the usual $M/G/1$ queue with infinite buffer, except for one detail: if the buffer becomes empty, the service process continues, so the buffer content (number of customers in the system) can become *negative*; in fact, we allow it to become infinitely negative. Of course, this has no physical interpretation, but it is useful as a step towards studying the bounded $M/G/1/B$ system in the next section.

For this doubly-unbounded queue, we consider the state of the system at the beginning of service epochs. Because of the memoryless arrival process, these instants together form an *(embedded) Markov chain*. Let N_n (with $-\infty < N_n < \infty$) denote the buffer content at the nth embedded point, i.e., at the beginning of the nth service period ($n \geq 1$).

We define q_j as the probability of exactly j arrivals during one service interval. Therefore,

$$q_j = \int_0^\infty \frac{(\lambda s)^j}{j!} e^{-\lambda s} dF_X(s).$$

Next, we define $r_i^{(n)}$ as the probability that $N_n = i$, assuming that the first service starts in state 0. Furthermore, we define r_i as the expected number of times the Markov chain visits state i. Clearly,

$$r_i = \sum_{n=1}^{\infty} r_i^{(n)}.$$

Note that although r_i is the expected number of visits during an infinite interval, it is (in general) a finite number, because the system will eventually drift to either $-\infty$ (if $\rho < 1$) or $+\infty$ (if $\rho > 1$).

It is easily seen that $r_i^{(n)}$ must satisfy the following recursion for $n > 1$:

$$r_i^{(n)} = \sum_{j=0}^{\infty} q_j r_{i-j+1}^{(n-1)}, \tag{4.1}$$

with boundary condition at $n = 1$:

$$r_i^{(1)} = I_{i=0} \stackrel{\text{def.}}{=} \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

because we start at level 0. Suppose one were to start in state m instead of 0, which corresponds to replacing $I_{i=0}$ by $I_{i=m}$ in the above equation. Since the system is doubly-unbounded, the resulting solution r_i' would just be a translated copy of the original solution, i.e., $r_i' = r_{i-m}$.

We now define $V(z)$ as the z-transform of q_j; it can be expressed in terms of the Laplace-Stieltjes transform $\tilde{F}_X(\cdot)$ as follows:

$$V(z) = \sum_{j=0}^{\infty} z^j q_j = \tilde{F}_X(\lambda - \lambda z). \tag{4.3}$$

Below, we will also need the solutions of the equation

$$V(K) = K. \tag{4.4}$$

It is easily seen that $V(z)$ is a convex function, that $V(1) = 1$ (so 1 is a solution to (4.4)), and that $V'(1) = \rho$. Because of these facts, (4.4) can have at most one other solution, which must be greater than 1 if $\rho < 1$, and less than 1 if $\rho > 1$. For our analysis, we assume that this second solution of (4.4) does indeed exist[1], and we denote it by K $(\neq 1)$. Again because of convexity, $V'(K)$ must be greater than 1 if $\rho < 1$, and less than 1 if $\rho > 1$. We denote the two solutions of (4.4) by K_1 and K_2, where $0 < K_1 < K_2$; Table 4.1 summarizes their properties.

In Appendix 4.A we prove the following theorem:

Theorem 4.1 *Given the recursion (4.1) with initial condition (4.2), the sum* $r_i = \sum_{n=1}^{\infty} r_i^{(n)}$ *(which can be interpreted as the expected number of visits to state i of*

[1]For $\rho < 1$, such a solution $K > 1$ obviously only exists if the Laplace transform $\tilde{F}_X(\cdot)$ exists for negative values of its argument. If the tail of the probability distribution of the service time X decays less than exponentially fast, this is a problem. In particular, K does not exist for distributions with a heavy tail, so the results in this chapter are not applicable in such cases.

case	K_1	$V'(K_1)$	K_2	$V'(K_2)$
$\rho < 1$	1	ρ	$K > 1$	> 1
$\rho > 1$	$K < 1$	< 1	1	ρ

Table 4.1: Properties of K_1 and K_2, solutions of $V(K) = K$.

the embedded Markov chain of the doubly-unbounded system) has the following properties:

$$r_i = \frac{K_1^{-i}}{1 - V'(K_1)} \quad \textit{for } i \leq 0, \tag{4.5}$$

and

$$\lim_{i\to\infty} K_2^i r_i = \frac{1}{V'(K_2) - 1}, \tag{4.6}$$

where $0 < K_1 < K_2$ *are the two solutions of* $V(K) = K$.

Note that (4.6) can also be written as

$$r_i \approx \frac{K_2^{-i}}{V'(K_2) - 1} \quad \text{for } i \gg 0. \tag{4.7}$$

4.3 The bounded $M/G/1$ queue

Let us now turn to the "real" system, the bounded $M/G/1/B$ queue. Because of the Poisson arrival process, we can again define an embedded Markov chain with embedding points at the beginning of service epochs. At those embedded points, the state variable of interest is the number of customers in the system. Starting in state A (i.e., with A customers in the system, and at the beginning of a service period), we study the evolution of the embedded Markov chain until absorption, which happens in either of two ways:

- Full buffer: if during one service, so many arrivals occur that there would be B or more customers in the system just before the completion of this service, full buffer is reached.

- Empty system: if there is only one customer left in the system, and during his service no others arrive, the system would be empty at the completion of this service.

We will now proceed to determine the expected number of times E_i the embedded Markov chain visits state i, starting from level A and ending in one of the above two absorbing states.

In the previous section, we have determined r_i for the doubly-unbounded queue starting in state 0. Those results can be used to obtain E_i as follows:

if one would just use the r_i results (shifted by A, to accommodate the fact that we start in state A instead of 0), one would overestimate E_i. In order to compensate for this error, we compare the expected number of times state i is visited in the bounded system (E_i) and in the unbounded system ($E'_i = r_{i-A}$):

- First, we have a contribution which is the same for both E_i and E'_i, corresponding to the evolution up to absorption (i.e., full buffer or empty system).
- Second, if the systems reach level 0 before level B, the bounded system stops (empty system), whereas the unbounded system continues, giving some additional contribution to E'_i. This is exactly as large as the contribution that would be produced by starting from state 0, which we know is given by r_i. In order to cancel this contribution, we need to determine the probability α that the unbounded system indeed reaches level 0 before level B. Then the correction term for E_i is clearly given by $-\alpha r_i$.
- Third, if the systems reach level B before level 0, the bounded system stops (full buffer), whereas the unbounded system continues, giving an additional contribution to E'_i (for $i < B$) only if it down-crosses into state $B-1$ later on. This contribution is exactly as large as the contribution that would be produced by starting from state $B-1$, which is given by r_{i-B+1}. In order to cancel this contribution, we need to determine the probability β that the unbounded system down-crosses into level $B-1$ before reaching level 0. Then the correction term is given by $-\beta r_{i-B+1}$.
  Note that if $\rho > 1$, the system may not return to level $B-1$ after having passed level B; in this case β is not equal to $1-\alpha$. On the other hand, if $\rho < 1$, then $\beta = 1-\alpha$.

Figure 4.1 shows four typical sample paths of the number of customers in the buffer as a function of time in the unbounded system. The filled circles represent the embedding points of the embedded Markov chain. The lines at levels 0 and B represent the absorption of the bounded system at empty system and full buffer, respectively. The dotted parts of the paths are the parts that must be compensated for by the above procedure. Note the difference between what happens to paths that reach level B and to paths that reach level 0. In the former case, compensation is necessary only if the buffer content returns to level $B-1$ (which may never happen if the arrival rate is higher than the service rate, i.e., $\rho > 1$).

From the above, it follows that E_i is given by r_{i-A} (that is, the expected number of visits to level i starting from level A in the doubly-unbounded system), minus the contribution due to sample paths beyond absorption at level 0 (i.e., αr_i) and level B (i.e., βr_{i-B+1}):

$$E_i = r_{i-A} - \alpha r_i - \beta r_{i-B+1}. \tag{4.8}$$

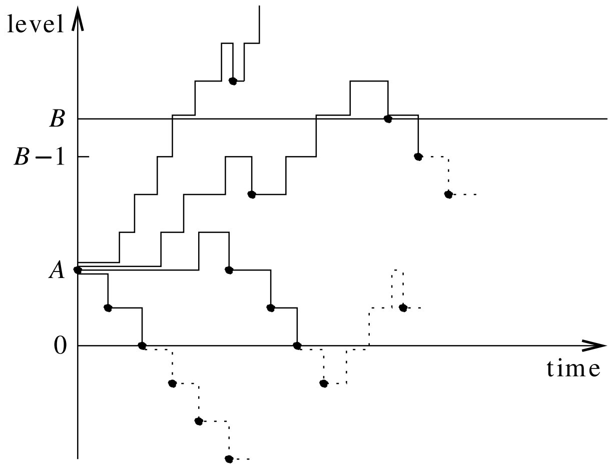


Figure 4.1: Illustration of typical sample paths in the doubly-unbounded queue. In the bounded queue, the dotted parts of the sample paths must be cancelled.

The value of the starting level A is determined by whether first or subsequent hits are being considered. The values of α and β can be determined by applying the appropriate boundary conditions, as will be shown in the sequel.

4.3.1 First hit: $A = 1$

In the case of first hit, we start in state 1, thus $A = 1$. In the bounded system, the embedded Markov chain cannot reach state 0 or $B - 1$ because of absorption (which would occur before or upon entering either state). As we also do not start in either of these states, we know that $E_0 = 0$ and $E_{B-1} = 0$. By inserting this into (4.8), we find the conditions for α and β:

$$0 = E_0 = r_{-1} - \alpha r_0 - \beta r_{-B+1} \tag{4.9}$$

and

$$0 = E_{B-1} = r_{B-2} - \alpha r_{B-1} - \beta r_0. \tag{4.10}$$

Substituting for r_i from (4.5) and (4.7) we get the following two equations:

$$K_1 - \alpha - \beta K_1^{B-1} = 0$$

and

$$\frac{K_2^{-(B-1)}(K_2 - \alpha)}{V'(K_2) - 1} \approx \frac{\beta}{1 - V'(K_1)}.$$

By substituting from (4.5) and (4.7) into (4.8), and then using the above equation to eliminate β, we can write E_i for $1 \ll i \leq B$ as follows:

$$\begin{aligned} E_i &\approx \frac{K_2^{-i+1}}{V'(K_2)-1} - \alpha\frac{K_2^{-i}}{V'(K_2)-1} - \beta\frac{K_1^{-i+B-1}}{1-V'(K_1)} \\ &\approx \frac{K_2^{-i}(K_2-\alpha)}{V'(K_2)-1} - \frac{K_1^{-i+B-1}K_2^{-(B-1)}(K_2-\alpha)}{V'(K_2)-1} \\ &= \frac{K_2^{-(B-1)}(K_2-\alpha)}{V'(K_2)-1}(K_2^{B-i-1} - K_1^{B-i-1}). \end{aligned}$$

Writing this properly as a limit gives us

Theorem 4.2 *The expected number of visits E_{B-j} to state $B-j$ of the $M/G/1/B$ embedded Markov chain, starting from state 1, has the following asymptotic behaviour for large B:*

$$\lim_{B\to\infty} \frac{E_{B-j}}{K_2^{-(B-1)}(K_2^{j-1} - K_1^{j-1})} = \frac{K_2-\alpha}{V'(K_2)-1}. \tag{4.11}$$

For the moment, we do not need the value of α and defer its calculation to Section 4.6.

4.3.2 Subsequent hits: $A = B - 1$

In the case of subsequent hits, we start in state $B-1$, thus $A = B-1$. In the bounded system, the embedded Markov chain cannot reach this state again (because of absorption), so the total number of visits to this state must be 1, i.e., $E_{B-1} = 1$. Furthermore, since state 0 is an absorbing state, it is considered unreachable, so $E_0 = 0$. By inserting this into (4.8), we find the equations for determining α and β:

$$1 = E_{B-1} = r_0 - \alpha r_{B-1} - \beta r_0$$

and

$$0 = E_0 = r_{-B+1} - \alpha r_0 - \beta r_{-B+1}.$$

Substituting for r_i from (4.5) and (4.7) in the above, we get

$$-\alpha\frac{K_2^{-(B-1)}}{V'(K_2)-1} + (1-\beta)\frac{1}{1-V'(K_1)} \approx 1$$

and

$$-\alpha + (1-\beta)K_1^{B-1} = 0.$$

Solving for α and β yields

$$\alpha = (1-\beta)K_1^{B-1} \tag{4.12}$$

and

$$\frac{1}{1-\beta} \approx -\frac{K_1^{B-1} K_2^{-(B-1)}}{V'(K_2)-1} + \frac{1}{1-V'(K_1)}. \tag{4.13}$$

By substitution into (4.8), we find for $1 \ll i \leq B$

$$E_i \approx \frac{\frac{K_1^{-i+B-1}}{1-V'(K_1)} - \frac{K_1^{B-1} K_2^{-i}}{V'(K_2)-1}}{\frac{1}{1-V'(K_1)} - \frac{K_1^{B-1} K_2^{-(B-1)}}{V'(K_2)-1}}.$$

Note that because $K_1/K_2 < 1$ for any $\rho \neq 1$, the second terms of both the numerator and the denominator vanish for large B and large i, which yields

Theorem 4.3 *The expected number of visits E_{B-j} to state $B-j$ of the $M/G/1/B$ embedded Markov chain, starting from state $B-1$, has the following asymptotic behaviour for large B:*

$$\lim_{B\to\infty} \frac{E_{B-j}}{K_1^{j-1}} = 1, \tag{4.14}$$

which for $\rho < 1$ reduces to

$$\lim_{B\to\infty} E_{B-j} = 1. \tag{4.15}$$

Remark 4.1 Here we give a less rigorous, but more intuitive explanation of (4.15).

Consider the embedded Markov chain of the bounded system at service beginning epochs, in the limit for $B \to \infty$. The probability of going from state m to state $m-1$ is simply equal to the probability of no arrivals during a service period, which we henceforth denote by γ. Starting in state m, define $\bar{E}_m$ to be the expected number of visits to state m before reaching any state above m. Clearly, $\bar{E}_m \geq 1$, since the starting in state m is also counted. Furthermore, for infinitely high levels $\bar{E}_m$ is independent of m, so $\bar{E}_m$ is equal to some constant E. Since we are considering subsequent hits, E_m (as defined earlier) is the expected number of visits to state m, starting from state $B-1$ until absorption due to a full buffer or an empty system. One can easily see that it must satisfy the recursion $E_m = \gamma E_{m+1}\bar{E}_m$, which for sufficiently large m reduces to

$$E_m \approx C E_{m+1}$$

with $C = \gamma E$. Since the starting state $B-1$ is never visited again until absorption, $E_{B-1} = 1$ is a boundary condition for the above recursion. It follows that

$$E_m \approx C^{B-m-1} \quad \text{for } 1 \ll m \leq B-1.$$

To determine C, we use the following two arguments. Since the embedded Markov chain eventually reaches an absorbing state at full buffer or empty system, E_m must be bounded for all m and B (with $m < B$), which is possible only if $C \leq 1$. On the other hand, since $\rho < 1$, there is a non-zero probability that the embedded Markov chain eventually reaches state 0. Therefore, E_m must not vanish for low values of m even at large B, which is possible only if $C \geq 1$. Obviously, the only value of C which satisfies both conditions is $C = 1$. Consequently, $E_m \approx 1$, for $1 \ll m \leq B - 1$.

4.4 Past and remaining service time distributions

Denote by X_n the duration of the service that starts at the nth embedded point ($n \geq 1$). As for the doubly-unbounded system, N_n is the state (number of customers) of the system at the nth embedded point. Without loss of generality, we assume that after absorption (due to either full buffer or empty system) the embedded Markov chain enters state 0 and stays there, i.e., N_n becomes 0. Define S_n to be the time, starting from the nth embedded point, until full buffer would be reached in the absence of any further service completions. Clearly, S_n has an Erlang-$(B - N_n)$ distribution, whose density for a given $N_n = i$ we denote by $g_i(s)$; thus

$$g_i(s) = f_{S_n}(s \mid N_n = i) = \lambda \frac{(\lambda s)^{B-i-1}}{(B-i-1)!} e^{-\lambda s}.$$

Write the past service time distribution as a sum over a set of disjoint events, which together cover all ways the event $Y \leq y$ can happen:

$$F_Y(y) = \mathbb{P}(Y \leq y) = \sum_{n=1}^{\infty} \sum_{i=1}^{B-1} \mathbb{P}(S_n \leq y \wedge S_n \leq X_n \wedge N_n = i).$$

Note that if $S_n \leq X_n$, then n is the last embedded point before reaching full buffer, in which case $Y = S_n$. Furthermore, the second summation is over the non-absorbing states $1 \leq i \leq B - 1$, thus restricting the first summation to embedded points until absorption. Next, conditioning on $N_n = i$ gives

$$F_Y(y) = \sum_{n=1}^{\infty} \sum_{i=1}^{B-1} \mathbb{P}(S_n \leq y \wedge S_n \leq X_n \mid N_n = i)\, \mathbb{P}(N_n = i).$$

Using the independence of S_n and X_n, and $\sum_{n=1}^{\infty} \mathbb{P}(N_n = i) = E_i$, we find:

$$\begin{aligned} F_Y(y) &= \sum_{i=1}^{B-1} E_i \int_0^{\infty} \int_0^{\infty} 1_{s \leq y}\, 1_{s \leq x}\, dF_X(x)\ g_i(s)\, ds \\ &= \int_0^{y} \bar{F}_X(s) H(s)\, ds, \end{aligned}$$

where $H(s)$ is defined as

$$H(s) = \sum_{i=1}^{B-1} g_i(s)\, E_i = \lambda \sum_{i=0}^{B-2} \frac{(\lambda s)^i}{i!} e^{-\lambda s}\, E_{B-i-1}. \tag{4.16}$$

By differentiation, one finds the probability density of Y:

$$f_Y(y) = \frac{dF_Y(y)}{dy} = H(y)\bar{F}_X(y), \tag{4.17}$$

which holds only if $\bar{F}_X(x)$ is continuous at $x = y$. At a discontinuity of $\bar{F}_X(\cdot)$, f_Y does not exist.

Similarly we can write for the remaining service time distribution upon reaching full buffer

$$\begin{aligned}
\mathbb{P}(z < Z < \infty) &= \sum_{n=1}^{\infty}\sum_{i=1}^{B-1} \mathbb{P}(X_n - S_n > z \mid N_n = i)\ \mathbb{P}(N_n = i) \\
&= \sum_{i=1}^{B-1} E_i \int_0^\infty \int_0^\infty 1_{x-s>z}\, dF_X(x)\ g_i(s)\, ds \\
&= \int_0^\infty \bar{F}_X(z+s) H(s)\, ds.
\end{aligned}$$

Differentiation yields the probability density:

$$\begin{aligned}
f_Z(z) &= -\frac{d\,\mathbb{P}(Z > z)}{dz} = -\frac{d}{dz}\int_z^\infty \bar{F}_X(t) H(t-z) dt \\
&= \bar{F}_X(z) H(0) + \int_{t=z}^{t=\infty} \bar{F}_X(t)\, dH(t-z) \\
&= \bar{F}_X(z) H(0) - \bar{F}_X(z) H(0) - \int_z^\infty H(t-z)\, d\bar{F}_X(t) \\
&= \int_z^\infty H(t-z)\, dF_X(t).
\end{aligned} \tag{4.18}$$

Just like f_Y, also f_Z does not exist at discontinuities of $\bar{F}_X(\cdot)$.

4.4.1 First hit

For the first hit and $\rho \neq 1$, E_i is given by (4.11), and using (4.16) we find the asymptotic expression for $H(s)$:

$$\begin{aligned}
H(s) &\approx \frac{K_2^{-(B-1)}\lambda(K_2-\alpha)}{V'(K_2)-1}\left(\sum_{i=0}^{B-2}\frac{(\lambda s K_2)^i}{i!}e^{-\lambda s} - \sum_{i=0}^{B-2}\frac{(\lambda s K_1)^i}{i!}e^{-\lambda s}\right) \\
&\approx \frac{K_2^{-(B-1)}\lambda(K_2-\alpha)}{V'(K_2)-1}\left(\sum_{i=0}^{\infty}\frac{(\lambda s K_2)^i}{i!}e^{-\lambda s} - \sum_{i=0}^{\infty}\frac{(\lambda s K_1)^i}{i!}e^{-\lambda s}\right) \\
&= \frac{K_2^{-(B-1)}\lambda(K_2-\alpha)}{V'(K_2)-1}\left(e^{\lambda(K_2-1)s} - e^{\lambda(K_1-1)s}\right).
\end{aligned}$$

According to (4.17), we find the probability density of the past service time upon reaching full buffer by multiplying $H(y)$ by $\bar{F}_X(y)$. For $\rho < 1$ we have $K_1 = 1$, so this can be simplified to

$$f_Y(y) \approx \frac{K_2^{-(B-1)}\lambda(K_2 - \alpha)}{V'(K_2) - 1}\left(e^{\lambda(K_2-1)y} - 1\right)\bar{F}_X(y),$$

and for $\rho > 1$, we have $K_2 = 1$, yielding

$$f_Y(y) \approx \frac{\lambda(1-\alpha)}{\rho - 1}\left(1 - e^{\lambda(K_1-1)y}\right)\bar{F}_X(y).$$

The above distributions are, in general, defective. If only the *conditional* distribution of the remaining service times is of interest (i.e., conditional on reaching full buffer), the above expressions and (4.18) can easily be normalized, leading to

Theorem 4.4 *The probability densities of the past and remaining service times in an $M/G/1/B$ queue, conditional on reaching full buffer, and starting from empty system, have the following asymptotic form:*

$$\lim_{B\to\infty} f_{Y|\text{fb}}(y) = \frac{\lambda}{1-\rho}(e^{\lambda(K-1)y} - 1)\bar{F}_X(y) \tag{4.19}$$

and

$$\lim_{B\to\infty} f_{Z|\text{fb}}(z) = \frac{\lambda}{1-\rho}\int_z^\infty \left(e^{\lambda(K-1)(t-z)} - 1\right) dF_X(t), \tag{4.20}$$

with $K = K_2$ if $\rho < 1$ and $K = K_1$ if $\rho > 1$.

4.4.2 Subsequent hits

For subsequent hits and $\rho \neq 1$, E_i is given by (4.14). As in the previous section, $H(s)$ can be found using (4.16), yielding

$$H(s) \approx \lambda e^{\lambda(K_1-1)s},$$

so according to (4.17) the past service time density is

$$f_Y(y) \approx \lambda e^{\lambda(K_1-1)y}\bar{F}_X(y), \tag{4.21}$$

and the remaining service time density is

$$f_Z(z) \approx \lambda\int_z^\infty e^{\lambda(K_1-1)(t-z)} dF_X(t). \tag{4.22}$$

Note that for $\rho > 1$, these distributions are not defective. For $\rho < 1$ we have $K_1 = 1$, which reduces the above expressions to

$$f_Y(y) \approx \lambda\bar{F}_X(y),$$
$$f_Z(z) \approx \lambda\bar{F}_X(z).$$

These are defective distributions. Their total probability is easily shown to be ρ, allowing us to calculate the densities conditional on reaching full buffer.

Theorem 4.5 *The probability densities of the past and remaining service times in an $M/G/1/B$ queue, conditional on reaching full buffer, and starting from full buffer, have the following asymptotic form:*

$$\lim_{B\to\infty} f_{Y|\text{fb}}(y) = \begin{cases} \frac{\lambda}{\rho}\bar{F}_X(y) & \text{for } \rho < 1, \\ \lambda e^{\lambda(K_1-1)y}\bar{F}_X(y) & \text{for } \rho > 1 \end{cases}$$

and

$$\lim_{B\to\infty} f_{Z|\text{fb}}(z) = \begin{cases} \frac{\lambda}{\rho}\bar{F}_X(z) & \text{for } \rho < 1, \\ \lambda \int_z^\infty e^{\lambda(K_1-1)(t-z)} dF_X(t) & \text{for } \rho > 1. \end{cases}$$

4.5 The limit $\rho \to 1$

For $\rho = 1$, equation (4.4) has only one solution (K_1 and K_2 approach 1 as ρ approaches 1). Since the analysis so far assumes two distinct solutions of (4.4), the obtained results may not hold for $\rho = 1$. However, we show that the limits of these results as $\rho \uparrow 1$ and as $\rho \downarrow 1$ exist and are identical, so we can assume them to be the result for $\rho = 1$.

4.5.1 First hit

In order to calculate the limit for the first hit, we need to examine the behaviour of K for ρ near 1. Consider the function

$$g(\rho, z) = \begin{cases} \frac{V(z)-z}{1-z} & \text{for } z \neq 1, \\ 1-\rho & \text{for } z = 1, \end{cases}$$

where $V(z)$ is given in (4.3). The function $g(\rho, z)$ as defined above is continuous at $z = 1$, because $\lim_{z\to 1} g(\rho, z) = 1 - \rho$ (using L'Hospital's rule).
Clearly, for $\rho \neq 1$, the solution K of the equation $g(\rho, K) = 0$, is the same $K \neq 1$ which is defined in Section 4.2 as a solution of (4.4). Note that for $\rho \to 1$, also $K \to 1$. Calculation shows that for all ρ

$$\left.\frac{\partial\, g(\rho, z)}{\partial z}\right|_{z=1} = -\frac{\lambda^2 \mathbb{E}(X^2)}{2} \quad \text{and} \quad \left.\frac{\partial\, g(\rho, z)}{\partial \rho}\right|_{z=1} = -1.$$

Then the implicit function theorem applied to $g(\rho, z)$ at $z = 1$ and $\rho = 1$ implies that

$$\lim_{\rho\to 1} \frac{dK}{d\rho} = -\left.\frac{\partial\, g(\rho, z)/\partial \rho}{\partial\, g(\rho, z)/\partial z}\right|_{z=1} = -\frac{2}{\lambda^2 \mathbb{E}(X^2)}.$$

Using L'Hospital's rule, we get the following limit:

$$\lim_{\rho\to 1}\frac{\lambda}{1-\rho}\left(e^{\lambda(K-1)y}-1\right)=\lim_{\rho\to 1}\lambda\frac{\frac{d}{d\rho}\left(e^{\lambda(K-1)y}-1\right)}{\frac{d}{d\rho}(1-\rho)}=-\lambda\frac{d}{d\rho}(\lambda(K-1)y)\bigg]_{\rho=1}=\frac{2y}{\mathbb{E}(X^2)},$$

which we substitute into (4.19) to find the conditional probability density of the past service time for the first hit:

$$\lim_{\rho\to 1} f_{Y|\mathrm{fb}}(y)\approx\frac{2y}{\mathbb{E}(X^2)}\bar{F}_X(y).$$

Similarly, the limit of the conditional remaining service time distribution for the first hit (as given by (4.20)) is

$$\lim_{\rho\to 1} f_{Z|\mathrm{fb}}(z)\approx\frac{2}{\mathbb{E}(X^2)}\int_z^\infty (t-z)dF_X(t).$$

4.5.2 Subsequent hits

To get the past service time distribution for subsequent hits when $\rho=1$, we need to calculate the limit of (4.21) and (4.22) for $\rho\to 1$. These limits are trivial, since these functions turn out to be continuous at $\rho=1$. So all results derived in Section 4.4.2 are also valid for $\rho=1$.

4.6 Approximation for full-buffer probability

In Section 4.4, we found expressions for the asymptotic distributions of the past and remaining service times upon reaching a high level (e.g., full buffer in the bounded system). It was noted that these distributions are defective; i.e., the total probability of these distributions is less than 1. This defect of course represents the fact that the system does not always reach full buffer.

In Section 4.3 we defined α to be the probability that the bounded system hits level 0 before reaching full buffer. As any path must either be absorbed at 0 or at B, we can conclude that the probability of reaching full buffer (i.e., absorption at B) is given by $1-\alpha$, which we calculate in the following.

4.6.1 First hit

Eliminating β from (4.9) and (4.10) in Section 4.3.1 gives:

$$\frac{K_2^{-(B-1)}(K_2-\alpha)}{V'(K_2)-1}\approx\frac{K_1^{-(B-1)}(K_1-\alpha)}{1-V'(K_1)}.$$

For both $\rho < 1$ ($K = K_2$) and $\rho > 1$ ($K = K_1$), this is

$$\frac{K^{-(B-1)}(K-\alpha)}{V'(K)-1} \approx \frac{1-\alpha}{1-\rho},$$

so

$$1-\alpha \approx \frac{K^{-(B-1)}(K-1)}{\frac{V'(K)-1}{1-\rho} - K^{-(B-1)}},$$

which for large B approaches

$$1-\alpha \approx \begin{cases} \frac{K(K-1)(1-\rho)}{V'(K)-1}K^{-B} & \text{for } \rho < 1, \\ 1-K & \text{for } \rho > 1. \end{cases} \tag{4.23}$$

Remark 4.2 Relationship with large deviation results

For $\rho < 1$, the decay rate of the full-buffer probability in (4.23) is given by $\log K$, where K is determined from (4.4). For verification, we will now show that this is equal to the decay rate obtained using large deviation theory.

According to [Sad91], the large-deviations calculation of the decay rate of the full-buffer probability starts by finding the non-trivial solution θ^* of the equation $\tilde{F}_X(-\theta^*) = \frac{\lambda+\theta^*}{\lambda}$. Then the decay rate is given by $\log K'$, with $K' = \frac{\lambda+\theta^*}{\lambda}$. Using the latter equality to rewrite the former in terms of K', we get $\tilde{F}_X(\lambda - \lambda K') = K'$. This is equivalent to (4.4), so $K' = K$.

Finally, it is interesting to note that $V'(K)$ can easily be shown to be the traffic intensity (i.e., average arrival rate divided by average service rate) in the so-called "θ^*-conjugate" system, in which the inter-arrival and service time distributions are exponentially twisted with the parameters θ^* and $-\theta^*$, respectively [Sad91].

4.6.2 Subsequent hits

For calculating the full-buffer probability for subsequent hits, we start from equations (4.12) and (4.13). By substituting the latter into the former, we obtain:

$$\alpha \approx \frac{K_1^{B-1}}{-\frac{K_1^{B-1}K_2^{-(B-1)}}{V'(K_2)-1} + \frac{1}{1-V'(K_1)}}.$$

This can be simplified by considering the cases $\rho < 1$ and $\rho > 1$ separately. First the case $\rho < 1$, which implies that $K_1 = 1$ and $K = K_2$:

$$1-\alpha \approx 1 - \frac{1}{-\frac{K^{-(B-1)}}{V'(K)-1} + \frac{1}{1-\rho}} \approx \rho, \tag{4.24}$$

where the second step uses the fact that B is large and $K > 1$. For the case $\rho > 1$, which implies that $K = K_1$ and $K_2 = 1$, we find:

$$1-\alpha \approx 1 - \frac{1}{-\frac{1}{\rho-1} + \frac{K^{-(B-1)}}{1-V'(K)}} \approx 1-(1-V'(K))K^{B-1}, \tag{4.25}$$

case	ρ	B	approximation	exact	difference
first hit	0.8	5	$8.327 \cdot 10^{-2}$	$9.851 \cdot 10^{-2}$	15 %
		10	$9.659 \cdot 10^{-3}$	$9.835 \cdot 10^{-3}$	1.8 %
		20	$1.2996 \cdot 10^{-4}$	$1.2999 \cdot 10^{-4}$	0.024 %
		40	$2.3526 \cdot 10^{-8}$	$2.3526 \cdot 10^{-8}$	0.000004%
	0.95	5	0.06891	0.1933	64 %
		10	0.04144	0.06759	39 %
		20	0.01498	0.01742	14 %
		40	0.001959	0.001995	1.8 %
	1.05	5	0.09370	0.2700	65 %
		10	0.09370	0.1560	40 %
		20	0.09370	0.1101	15 %
		40	0.09370	0.09570	2.1 %
	1.2	5	0.3137	0.3900	20 %
		10	0.3137	0.3233	3.0 %
		20	0.3137	0.3139	0.069 %
		40	0.3137	0.3137	0.000035 %
subsequent hit	0.95	5	0.95	0.8597	11 %
		10	0.95	0.9184	3.4 %
		20	0.95	0.9419	0.9 %
		40	0.95	0.9491	0.09 %
	1.05	5	0.9674	0.9060	6.8 %
		10	0.9800	0.9668	1.4 %
		20	0.9925	0.9912	0.13 %
		40	0.9990	0.9989	0.002 %

Table 4.2: Comparison of approximation and true values for the probability of reaching full buffer in an $M/D/1/B$ queue.

where the second step uses the fact that B is large and $K < 1$. From this, one sees that the full-buffer probability $1 - \alpha \approx 1$, which is not surprising, since we start from just below full buffer in a system with a higher arrival rate than service rate ($\rho > 1$).

4.7 Numerical validations and an application

In the previous sections, we have derived several asymptotic results which are valid for infinitely high levels in $M/G/1$ queues. For sufficiently high levels, the results may still be used as approximations. In general, it is difficult to calculate error bounds for these approximations. However, we note that many

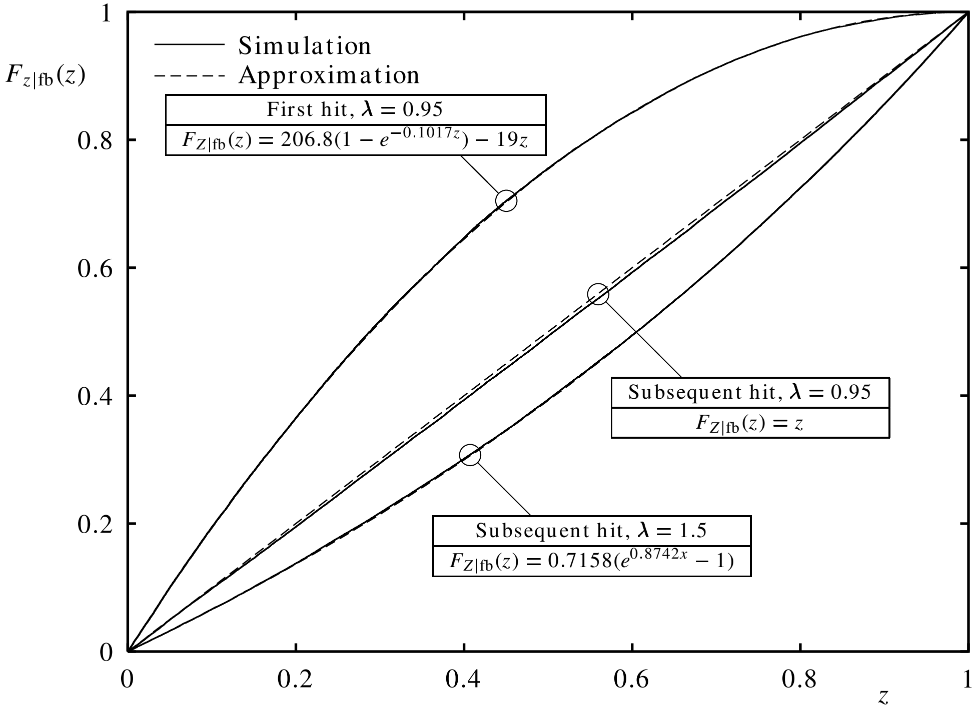


Figure 4.2: Remaining service time distributions in an $M/D/1/10$ queue.

of the approximations involve neglecting terms of the form K^B (if $K < 1$) or K^{-B} (if $K > 1$). For such a term to be very small, B must be very large and/or K must be far from 1. The value of K depends both on the form of the service time distribution $F_X(\cdot)$ and on ρ. If ρ approaches 1, K also approaches 1. Consequently, we can expect the approximation to be good if B is large and ρ is not close to 1.

4.7.1 Example: $M/D/1/B$

In order to illustrate the validity of the approximations, we first consider a simple $M/D/1/B$ queue. We assume the deterministic service time to be 1, thus $\mathbb{E}(X) = 1$, and $\rho = \lambda$. This leaves two parameters to vary, namely ρ (traffic intensity) and B (buffer size).

First, we will test our approximations for the full-buffer probability, presented in Section 4.6. For the $M/D/1/B$ queue, the full-buffer probability can be computed numerically (e.g., along the lines of Section 3.1). To validate our approximations, Table 4.2 shows the approximate and true values of the probability of reaching full buffer, starting from an empty system (i.e., first hit) and starting from level $B - 1$ (i.e., subsequent hits). As expected, the approximation is good for large B and for ρ not close to 1.

The accuracy of the approximations for the remaining service time distribution upon first and subsequent full-buffer hits is illustrated in Figure 4.2. (Note in this example that because of the deterministic service time of 1, the remaining service time cannot exceed 1.) The simulation results (shown with solid lines) of course have some small statistical errors, at most 0.004 with 95 % confidence. The analytical approximations (plotted with dashed lines) are given by the expressions shown in the figure, which follow directly from the analysis in Section 4.4. Clearly, the approximate analytical distributions agree quite well with the simulation results, especially considering the fact that we have a relatively small B and ρ close to 1.

4.7.2 Example: $M/H_2/1/B$

As an example with non-deterministic service time, we consider an $M/H_2/1/B$ queue. We choose the hyperexponential service time distribution such that the service rate is either 2 (with probability 1/2), or 2/3 (with probability 1/2); thus $\mathbb{E}(X) = 1$ and $\rho = \lambda$.

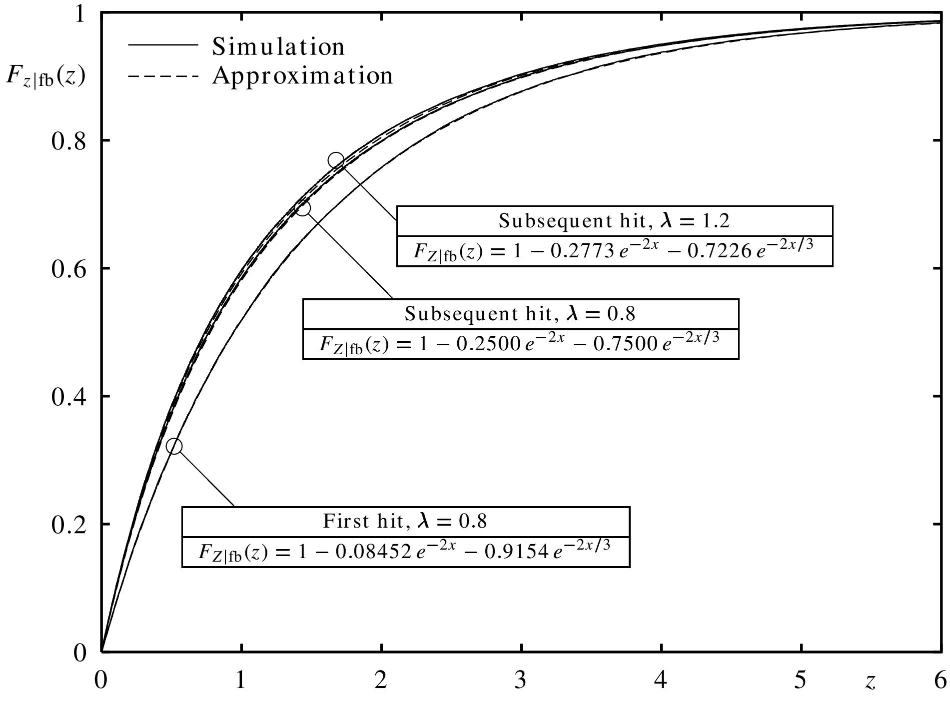


Figure 4.3: Remaining service time distributions in an $M/H_2/1/10$ queue.

case	ρ	B	approximation	exact	difference
first hit	0.4	10	0.0005005	0.0005007	0.054 %
		20	$3.098 \cdot 10^{-7}$	$3.098 \cdot 10^{-7}$	0.00 %
	0.6	10	0.00717	0.00728	1.5 %
		20	$1.213 \cdot 10^{-4}$	$1.213 \cdot 10^{-4}$	0.024 %
	0.8	10	0.03051	0.03618	16 %
		20	$5.145 \cdot 10^{-3}$	$5.284 \cdot 10^{-3}$	2.6 %
	1.2	10	0.1363	0.1744	22 %
		20	0.1363	0.1436	5.1 %
	1.6	10	0.3175	0.3238	1.9 %
		20	0.3175	0.3177	0.05 %
	2.0	10	0.4342	0.4355	0.3 %
		20	0.4342	0.4343	0.00 %
subsequent hit	0.4	10	0.4000	0.3997	0.07 %
		20	0.4000	0.4000	0.00 %
	0.6	10	0.6000	0.5942	0.97 %
		20	0.6000	0.5999	0.016 %
	0.8	10	0.8000	0.7629	4.9 %
		20	0.8000	0.7946	0.68 %
	1.2	10	0.9563	0.9441	1.3 %
		20	0.9899	0.9894	0.053 %
	1.6	10	0.9885	0.9882	0.03 %
		20	0.9997	0.9997	0.00 %
	2.0	10	0.9972	0.9972	0.00 %
		20	1.0000	1.0000	0.00 %

Table 4.3: Comparison of approximation and true values for the probability of reaching full buffer in an $M/H_2/1/B$ queue.

Let us first test the approximation for the full-buffer probability. Table 4.3 shows the results from our approximation, as well as results from a numerical computation for comparison. Clearly, for ρ not close to 1, our approximation is quite good. A comparison with Table 4.2 suggests that for the same ρ, the approximations are better for the $M/D/1/B$ system than for the $M/H_2/1/B$ system. Presumably, this is due to the larger variance of the service time in the latter system.

For the $M/H_2/1/B$ queue, Figure 4.3 shows the remaining service time distribution obtained from our analytical approximations and from simulations. Again, the agreement between our approximations and the simulation results is evident.

4.7.3 Application: estimation of consecutive cell loss probabilities in an $M/G/1/B$ queue

In Chapter 5, we consider the estimation of consecutive-cell-loss (CCL) probabilities in $M/G/1/B$ queues using importance sampling simulation. In the process of doing that, expression (5.4) is derived, which expresses the probability of at least one n-CCL event during a busy cycle in terms of four other probabilities. In Chapter 5, each of these four probabilities is estimated using simulation. Alternatively, these probabilities can also be approximated numerically using results from the present chapter, as demonstrated below.

Equation (5.4) expresses the n-CCL probability γ_n as follows:

$$\gamma_n = \gamma p_{1n} + \frac{\gamma(1 - p_{1n})\phi p_n}{1 - \phi(1 - p_n)}.$$

Two of the four probabilities involved, namely γ and ϕ, are simply the probabilities of reaching first and subsequent full-buffer periods, respectively, which can (for large B) be approximated by $1 - \alpha$ as calculated in Section 4.6 (equations (4.23) and (4.24)). The other two, p_{1n} and p_n, are the probabilities that at least n (Poisson) arrivals occur during a first and a subsequent full-buffer period, respectively. If the distributions of those durations are known, these two probabilities can be estimated using a straightforward integration:

$$\begin{aligned} p_{1n} &= \int_0^\infty e^{-\lambda x} \sum_{i=n}^{\infty} \frac{(\lambda x)^i}{i!}\, dG_1(x), \\ p_n &= \int_0^\infty e^{-\lambda x} \sum_{i=n}^{\infty} \frac{(\lambda x)^i}{i!}\, dG(x), \end{aligned} \tag{4.26}$$

where $G_1(\cdot)$ and $G(\cdot)$ are the distribution functions of the duration of first and subsequent full-buffer periods, respectively. If the overflow level B is high enough, $G_1(\cdot)$ and $G(\cdot)$ can be approximated by the asymptotic distributions that we have derived in the present chapter (Theorems 4.4 and 4.5).

As an example, consider an $M/D/1/B$ queue, with arrival rate 0.8 and deterministic service time $d = 1$. The approximate values for γ and ϕ can be read from Table 4.2. The duration of the first full-buffer period asymptotically has a density $dG_1(x)/dx$ of the form $\left(e^{0.43084(1-x)} - 1\right)$, while the distribution $G(x)$ of the duration of subsequent full-buffer periods asymptotically is uniform on $[0, 1]$. By numerical evaluation of the integrals in (4.26), approximate values of p_{1n} and p_n can be calculated. Finally, the four probabilities are substituted into (5.4) to obtain the n-CCL probability.

Clearly, the agreement between the analytical approximation and the exact results is very good; in fact, it is much better than should be expected. For example, consider $B = 5$: at such a low buffer size the approximations from the

B	n	γ_n (anal.appr.)	γ_n (exact)
5	1	$5.412 \cdot 10^{-2}$	$6.018 \cdot 10^{-2}$
	4	$7.239 \cdot 10^{-4}$	$7.246 \cdot 10^{-4}$
	16	$1.330 \cdot 10^{-17}$	$1.329 \cdot 10^{-17}$
	64	$1.174 \cdot 10^{-98}$	$1.175 \cdot 10^{-98}$
10	1	$6.278 \cdot 10^{-3}$	$6.352 \cdot 10^{-3}$
	4	$8.388 \cdot 10^{-5}$	$8.398 \cdot 10^{-5}$
	16	$1.542 \cdot 10^{-18}$	$1.542 \cdot 10^{-18}$
	64	$1.362 \cdot 10^{-99}$	$1.363 \cdot 10^{-99}$
20	1	$8.447 \cdot 10^{-5}$	$8.448 \cdot 10^{-5}$
	4	$1.130 \cdot 10^{-6}$	$1.130 \cdot 10^{-6}$
	16	$2.075 \cdot 10^{-20}$	$2.075 \cdot 10^{-20}$
	64	$1.832 \cdot 10^{-101}$	$1.834 \cdot 10^{-101}$

Table 4.4: Analytic approximation of n-CCL probability for $M/D/1/B$ queues, with $\lambda = 0.8$ and $d = 1$.

present chapter are generally rather bad, and indeed Table 4.2 lists an error of about 15% for the approximation of γ used here. Since γ_n is directly proportional to γ, a 15% error in γ should also contribute a 15% error to γ_n. At $n = 1$, γ_n indeed has an error of this order (11%), but at $n = 4$, 16 and 64, the error in γ_n is just 0.1%. It seems as if the large error in the approximation of γ is compensated for by errors in the approximations of ϕ, $dG_1(\cdot)$ and $dG(\cdot)$. Correlations between these four errors are of course to be expected, since they all come from one approximation method. However, it is surprising that they cancel so well; further analysis of this may be of interest.

4.8 Summary and concluding remarks

In this chapter, we have derived analytical approximations for the probability densities of the past and remaining service time upon reaching a high level, e.g., full buffer, in $M/G/1$ queues. Table 4.5 summarizes the main results for these distributions, conditional on reaching full buffer. As a by-product, we also obtained approximations for the probability of reaching full buffer (for the first and subsequent hits) in a busy cycle; those are given in (4.23), (4.24) and (4.25). However, the results in this chapter are only valid if a solution, unequal to 1, of (4.4) exists; in particular, this excludes cases where the distribution of the service time has a heavy tail.

Validations of the approximations are carried out by means of comparisons with true values obtained from exact numerical results and simulations. The approximations are shown to be most accurate for high levels and ρ not too close to 1, although the approximate distributions in Table 4.5 remain surprisingly accurate for ρ near 1.

An extension of our results to queueing systems other than $M/G/1$ would be of much interest. Our present analysis is based on an embedded Markov chain formulation, which cannot be applied to most $GI/G/1$ systems; for such systems, a different approach needs to be devised. The only other category of queueing systems which does lend itself to an embedded Markov chain analysis, is $GI/M/1$, for which a study of the remaining inter-arrival time at service completion epochs would be of interest.

Case		Past service time density $f_{Y\mid\text{fb}}(y)$	Remaining service time density $f_{Z\mid\text{fb}}(z)$
First hit	$\rho \neq 1$	$\frac{\lambda}{1-\rho}(e^{\lambda(K-1)y} - 1)\bar{F}_X(y)$	$\frac{\lambda}{1-\rho}\int_z^\infty \left(e^{\lambda(K-1)(t-z)} - 1\right) dF_X(t)$
	$\rho = 1$	$\frac{2}{\mathbb{E}(X^2)} y\bar{F}_X(y)$	$\frac{2}{\mathbb{E}(X^2)}\int_z^\infty (t-z)\, dF_X(t)$
Subs. hit	$\rho \leq 1$	$\frac{1}{\mathbb{E}(X)}\bar{F}_X(y)$	$\frac{1}{\mathbb{E}(X)}\bar{F}_X(z)$
	$\rho > 1$	$\lambda e^{\lambda(K-1)y}\bar{F}_X(y)$	$\lambda\int_z^\infty e^{\lambda(K_1-1)(t-z)}\, dF_X(t)$

Note: K is defined as the solution not equal to 1 of (4.4). For some distributions such a solution does not exist, and the above results are not valid.

Table 4.5: Probability densities of past and remaining service times upon reaching a high level in $M/G/1$ queues.

4.A Solving the doubly-unbounded system

In this appendix, we present a proof of Theorem 4.1. For definitions and properties of N_n, q_j, $V(\cdot)$, K_1 and K_2, see Section 4.2.

Define the following double-sided z-transform of the distribution of N_n:

$$R^{(n)}(z) = \sum_{i=-\infty}^{\infty} r_i^{(n)} z^i$$

From (4.2) we get

$$R^{(1)}(z) = 1,$$

and from (4.1) we have (for $0 < |z| \leq K_2$)

$$R^{(n)}(z) = \frac{V(z)}{z} R^{(n-1)}(z),$$

as can be seen either by expanding the summations, or by noting that N_n is just the sum of N_{n-1} and another independent random number with the generating function $V(z)/z$. Next, we easily find:

$$R^{(n)}(z) = R^{(1)}(z) \left(\frac{V(z)}{z}\right)^{n-1} = \left(\frac{V(z)}{z}\right)^{n-1}.$$

Now define $R(z)$ to be the double-sided z-transform of r_i, to find:

$$R(z) = \sum_{i=-\infty}^{\infty} r_i z^i = \sum_{i=-\infty}^{\infty} \sum_{n=1}^{\infty} r_i^{(n)} z^i = \sum_{n=1}^{\infty} R^{(n)}(z) = \sum_{n=1}^{\infty} \left(\frac{V(z)}{z}\right)^{n-1} = \frac{z}{z - V(z)}. \tag{4.27}$$

The change of the order of summation above is possible on the ring[2] $\{z : K_1 < |z| < K_2\}$ in the complex plane. So, (4.27) is valid on that ring.

Note that $R(z)$ itself is only defined on the ring; however, the right-hand side of (4.27) is also analytical outside the ring (except of course at K_1 and K_2), so it is the (unique) analytical continuation of $R(z)$.

Behaviour of r_i for $i \leq 0$. We have already shown that $|V(z)| < |z|$ on the ring $\{z : K_1 < |z| < K_2\}$. By Rouché's theorem (see, e.g., [Tit52]), this means that $V(z) - z$ has exactly as many zeros on the disc $\{|z| < K_2\}$ as z does, while the latter of course has exactly one zero (at 0). We already know that $V(z) - z$ has a zero at K_1, which is on the disc. So that must be its only zero. Consequently, $z = K_1$ is the only pole of $R(z)/z = 1/(z - V(z))$ on the disc. Now calculate the residue of this pole:

$$\begin{aligned}\operatorname{Res}\left(\frac{R(z)}{z}, K_1\right) &= \lim_{z\to K_1} (z - K_1)\frac{R(z)}{z} = \lim_{y\to 1}(y-1)K_1 \frac{R(K_1 y)}{K_1 y} \\ &= \lim_{y\to 1} \frac{(y-1)K_1}{K_1 y - V(K_1 y)} = \lim_{y\to 1} \frac{K_1}{K_1 - K_1 V'(K_1 y)} = \frac{1}{1 - V'(K_1)},\end{aligned}$$

where L'Hospital's rule was used in the fourth step. Knowing the only pole's residue, we can split $R(z)/z$ as follows (for $K_1 < |z| < K_2$):

$$\begin{aligned}\frac{R(z)}{z} &= \sum_{i=0}^{\infty} r'_{i+1} z^i + \frac{1}{1 - V'(K_1)} \frac{1}{z - K_1} = \sum_{i=0}^{\infty} r'_{i+1} z^i + \frac{1/K_1}{1 - V'(K_1)} \frac{K_1/z}{1 - K_1/z} \\ &= \sum_{i=0}^{\infty} r'_{i+1} z^i + \frac{1/K_1}{1 - V'(K_1)} \sum_{i=-\infty}^{-1} \left(\frac{z}{K_1}\right)^i.\end{aligned}$$

[2] Note that the last summation (a simple geometric sum) converges on the ring because there $|V(z)| < |z|$. This follows from observing that:

- Because of the convexity of $V(z)$ and the fact that $V(K_1) = K_1$ and $V(K_2) = K_2$, we have $V(z) < z$ for any (real and positive) z for which $K_1 < z < K_2$.
- Write z, which is in general a complex number, as $x + iy$. With the definition (4.3) of $V(z)$ in terms of a Laplace transform of a positive function, one easily verifies that $|V(z)| = |V(x+iy)| \leq |V(|x|)| \leq |V(|x + iy|)|$.

The reason the first term, $\sum_{i=0}^{\infty} r'_{i+1} z^i$, contains no negative powers of z, is that this term results from removing the only pole $R(z)$ has on $|z| < K_2$, so this term must be analytic on this disc, and therefore can be written as a Taylor series. Multiply the above by z to find

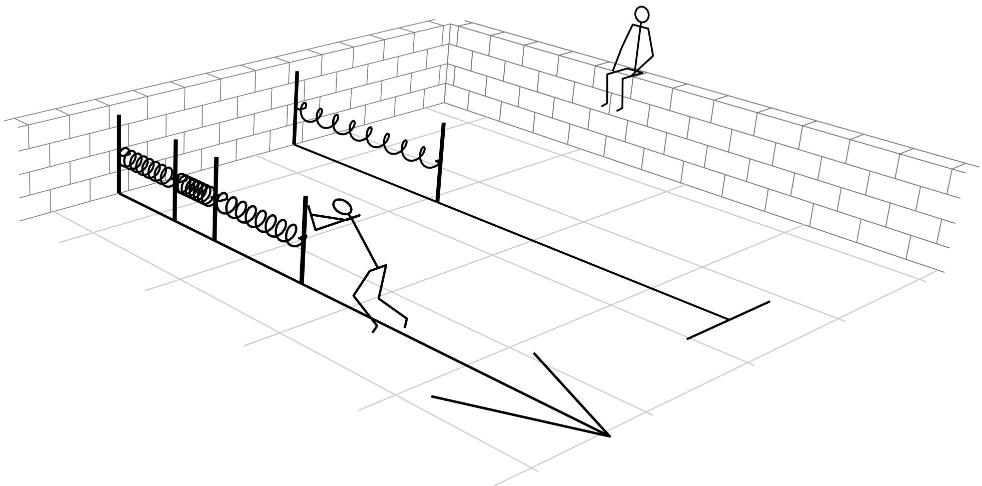
$$R(z) = \sum_{i=1}^{\infty} r'_i z^i + z \frac{1/K_1}{1 - V'(K_1)} \sum_{i=-\infty}^{-1} \left(\frac{z}{K_1}\right)^i = \sum_{i=1}^{\infty} r'_i z^i + \frac{1}{1 - V'(K_1)} \sum_{i=-\infty}^{0} \left(\frac{z}{K_1}\right)^i,$$

from which (4.5) directly follows.

Limit behaviour of r_i for large i. Using the final value theorem (Abelian theorem, see [Tit52]) for z-transforms and applying L'Hospital's rule yield the following limit:

$$\begin{aligned}\lim_{i\to\infty} K_2^i r_i &= \lim_{z\to 1} (1-z) R(K_2 z) = \lim_{z\to 1} \frac{(1-z)K_2 z}{K_2 z - V(K_2 z)} \\ &= \lim_{z\to 1} \frac{(1-z)K_2 - K_2 z}{K_2 - K_2 V'(K_2 z)} = \frac{1}{V'(K_2) - 1},\end{aligned}$$

thus proving (4.6). □



Chapter 5

Rare events involving IID sums: bounded threshold

The estimation of rare event probabilities involving sums of independent and identically distributed (i.i.d.) random variables is often encountered in the performance and reliability evaluation of models stemming from various applications, such as system reliability, quality of service in communication networks, signal detection, and others. Analytical and/or numerical evaluation of these probabilities may not be possible. One alternative would be simulation; however, this is not efficient unless some importance sampling procedure is devised to estimate these typically small probabilities.

Let $S_n = \sum_{i=1}^{n} X_i$ be the sum of n i.i.d. random variables. A lot is known about the asymptotic properties (for $n \to \infty$) of *averages* of i.i.d. random variables (i.e., S_n/n); for example, the central limit theorem (see, e.g., [Fel66]) and large deviation results (see, e.g., [SW95]). In this chapter, however, we look at the sum itself (not the average), which generally has different asymptotic properties for $n \to \infty$. For example, the probability that the average S_n/n is less than a given threshhold generally decreases exponentially with n (assuming the threshold is below the mean $\mathbb{E}X_i$). In contrast, the probability that the sum S_n is below a given threshhold often decreases even faster than exponentially with n (e.g., as fast as $1/n!$).

In this and the next chapter, we focus on the probability that the sum of n i.i.d. positive random variables X_i is less than another, independent and possibly differently distributed, positive random variable Y (the "threshold"):

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i < Y\right). \tag{5.1}$$

Our aim is to develop importance sampling simulation methods for the estimation of such probabilities. A desirable property of such simulation methods is asymptotic efficiency, which means (see, e.g., [Sad91]) that for a given relative error (accuracy), the required simulation effort grows less than exponential in n; or, equivalently, that for a given simulation effort, the relative error grows less than exponential in n. In this chapter, an importance sampling change of measure is proposed in which the distribution of Y is not modified, and which results in asymptotically efficient simulation if Y is upper-bounded.

In Section 5.1, the details of this change of measure are discussed, and the asymptotic efficiency is proved. In Section 5.2, this simulation scheme is applied to the estimation of consecutive loss probabilities in $M/G/1$ queues.

5.1 Importance sampling simulation

In this section, the change of measure is described, and its asymptotic efficiency is proved.

5.1.1 The change of measure

As stated in the introduction, only changes of measure are considered in which the distribution of Y is not changed; thus, only the distribution of the X_i remains to be changed. This may seem like an odd and artificial restriction, but it is not. In practical problems, like the one to be discussed in Section 5.2, the distribution of Y is not known because the samples of Y are simulation results themselves. In that case, it is not possible to directly change the distribution of Y. Furthermore, it is of theoretical interest to see what can be achieved by such a restricted change of measure, and compare this with the results for a less restricted change of measure that includes changing the distribution of Y, as discussed in Chapter 6.

For the change of measure applied to X_i, we only consider exponential tilting, since that often turns out to be asymptotically efficient (e.g., [Sad91]). Denote the exponential tilting parameter as θ, then the tilted density of X_i is given by $\rho e^{-\theta x} f_X(x)$ (for $x \geq 0$), where $f_X(x)$ is the original density of X_i and ρ is a normalization constant ($\rho^{-1} = \int_0^\infty e^{-\theta x} f_X(x) dx$).

In the following theorem we propose an exponential tilting parameter which yields an asymptotically efficient simulation as n goes to infinity.

Theorem 5.1 *Let Y be a random variable with an upper bound b and pdf $f_Y(\cdot)$, and let X_i be i.i.d. positive random variables with pdf $f_X(\cdot)$. Then, an asymptotically efficient change of measure for estimating the probability in (5.1) is obtained by exponentially tilting $f_X(\cdot)$, with a tilting parameter given by*

$$\theta = qn/b. \tag{5.2}$$

Here, q is defined such that the $(q-1)$-th derivative of $f_X(x)$ at $x = 0$ is its first non-zero derivative. Using this change of measure, the relative error asymptotically increases proportionally to $n^{v/2+3/4}$, with v defined such that the v-th derivative of $f_Y(y)$ at $y = b$ is its first non-zero derivative; if Y is deterministic (necessarily equal to b), we set $v = -1$.

Proof: See Section 5.1.2.

From the above theorem, it is clear that the optimal change of measure and the resulting error bounds depend in a rather peculiar way on the distributions of X_i and Y: the dependence on the X-distribution is only through that distribution's behaviour near zero, and the dependence on the Y-distribution is only through its behaviour near b. This is not surprising since it suggests that asymptotically (for large n), the typical way for the rare event to happen is by getting small values of X_i and a large value of Y.

5.1.2 Asymptotic efficiency

The proof of the asymptotic efficiency of the change of measure introduced above, relies strongly on an extension to the well-known central-limit theorem, given as Theorem 5.2 in Appendix 5.A.2. This theorem basically states that under some conditions, the distribution of the sum of a large number of i.i.d. random variables tends to a normal distribution also when the distribution of the individual random variables is exponentially tilted with a parameter that *varies* with the number of the i.i.d. variables.

Preliminaries

Let $g(x)$ be the density obtained by exponentially tilting the pdf $f_X(x)$ of X_i; then

$$g(x) = \rho e^{-\theta x} f_X(x),$$

where ρ is the normalization factor. Consider a simulation run in which we generate one sample y of Y, and n samples x_i of X_i, $i = 1, 2, \ldots, n$. Define the sum $s = \sum_{i=1}^{n} x_i$, and the indicator $I = 1_{s<y}$. The likelihood ratio associated with this simulation run is

$$L = \prod_{i=1}^{n} \rho^{-1} e^{\theta x_i} = \rho^{-n} e^{\theta s}.$$

The importance sampling estimate of the probability $\mathbb{P}\{\sum X_i < Y\}$ is given by $\mathbb{E}^*(LI)$, where $\mathbb{E}^*$ denotes the expectation w.r.t. the new probability measure. The variance of this estimator is given by $\mathbb{E}^*(L^2 I) - \mathbb{E}^{*2}(LI)$. Consequently, its relative error is given by

$$RE = \frac{\sqrt{\mathbb{E}^*(L^2 I) - \mathbb{E}^{*2}(LI)}}{\mathbb{E}(LI)} = \sqrt{\frac{\mathbb{E}^*(L^2 I)}{\mathbb{E}^{*2}(LI)} - 1},$$

so in order to prove the asymptotic efficiency, we need to upper bound the fraction in the right-hand side.

Proof of Theorem 5.1

The proof consists mainly of calculating the limit behaviour (as $n \to \infty$) of $\mathbb{E}^*(L^2 I)$ and $\mathbb{E}^*(LI)$. These limits can be calculated using the extended central limit theorem presented in Appendix 5.A.2. Since the calculations for $\mathbb{E}^*(L^2 I)$ and for $\mathbb{E}^*(LI)$ have a lot in common, it is convenient to first calculate the limit behaviour of the more general $\mathbb{E}^*(L^k I)$, and substitute $k = 1$ and $k = 2$ later. Evidently, $\mathbb{E}^*(L^k I)$ is given by the following integral, where $F_S(\cdot)$ is the probability distribution function of the sum of X_i drawn from their tilted distribution:

$$\mathbb{E}^*(L^k I) = \int_0^b \int_s^b L^k \, dF_Y(y) \, dF_S(s) = \rho^{-nk} \int_0^b \int_s^b e^{\theta s k} \, dF_Y(y) \, dF_S(s).$$

Apply the extended central limit theorem (Theorem 5.2) to approximate[1] $dF_S(s)$ as follows, noting that q as defined in Theorem 5.1 equals $r + 1$ with r as defined in Theorem 5.2:

$$\mathbb{E}^*(L^k I) \approx \rho^{-nk} \int_0^b \int_s^b dF_Y(y) e^{\theta s k} \mathfrak{n}_{nq/\theta, nq/\theta^2}(s) ds. \tag{5.3}$$

Here and in the following, the approximate-equals sign $\approx$ is used to denote that the limit as $n \to \infty$ of the ratio of the left-hand and right-hand side is 1. Since $\theta = qn/b$ increases proportional to n, the factor $e^{\theta s k}$ in the above integrand will become an ever steeper function near $s = b$ for large n. Therefore, in the limit as $n \to \infty$, the behaviour of the rest of the integrand is only important for s close to b.

Consider the inner integral in the above. Near b the density $f_Y(y)$ of Y is assumed (per the definition of v in the theorem; the special case of deterministic Y will be treated later) to have the form

$$C(b - y)^v,$$

[1]Note that due to the presence of another factor that depends on θ in the integrand, it is not immediately obvious that this gives a correct approximation of the integral. However, as we will see below the behaviour of the inner integral is such, that for large θ, only the behaviour of $dF_S(s)$ near its peak is important. Using this observation together with the extended central limit theorem, it can be shown that indeed the ratio of the left-hand and the right-hand sides of (5.3) goes to 1 as θ and n go to infinity.

for some positive constant C. Using this, the inner integral can be approximated as

$$\int_s^b C(b-y)^v dy = \frac{C}{v+1}(b-s)^{v+1}.$$

Substituting this into (5.3), and writing the Gaussian density $\mathfrak{n}$ explicitly, we find

$$\mathbb{E}^*(L^k I) \approx \frac{C\sqrt{nq}}{(v+1)\rho^{nk} b\sqrt{2\pi}} \int_0^b (b-s)^{v+1} e^{\theta sk} e^{-\frac{(b-s)^2 nq}{2b^2}} ds.$$

Next, substitute $s = b(1-z)$ and $\theta b = nq$:

$$\mathbb{E}^*(L^k I) \approx \frac{C\sqrt{nq} e^{nqk} b^{v+2}}{(v+1)\rho^{nk}\sqrt{2\pi}} \int_0^1 z^{v+1} e^{-nqkz} e^{-\frac{z^2 nq}{2}} dz.$$

For large n, the integrand goes to zero very quickly with increasing z, so we can approximate the integral as follows:

$$\begin{aligned}
\int_0^1 z^{v+1} e^{-nqkz} e^{-\frac{z^2 nq}{2}} dz &\approx \int_0^\infty z^{v+1} e^{-nqkz} e^{-\frac{z^2 nq}{2}} dz \\
&= (nq)^{-v-2} \int_0^\infty w^{v+1} e^{-kw} e^{-\frac{w^2}{2nq}} dw \\
&\approx (nq)^{-v-2} \int_0^\infty w^{v+1} e^{-kw} dw \\
&= (nq)^{-v-2} \frac{(v+1)!}{k^{v+2}},
\end{aligned}$$

where the substitution $w = nqz$ has been made. Thus

$$\mathbb{E}^*(L^k I) \approx \frac{Cv! e^{nqk} b^{v+2}}{(nq)^{v+3/2} k^{v+2} \rho^{nk} \sqrt{2\pi}}.$$

Substitution of $k = 1$ and $k = 2$ into the above yields:

$$\frac{\mathbb{E}^*(L^2 I)}{\mathbb{E}^{*2}(LI)} \approx \frac{\sqrt{2\pi}}{2^{v+2} Cv! b^{v+2}} (nq)^{v+3/2}.$$

We see that $\frac{\mathbb{E}^*(L^2 I)}{\mathbb{E}^{*2}(LI)}$ is proportional to $(nq)^{v+3/2}$. Consequently, the relative error of the estimator asymptotically increases proportional to $n^{v/2+3/4}$ for large n, as was to be shown.

Finally, consider the special case of deterministic Y. In that case, the inner integral in (5.3) is equal to 1, independent of s since all of the probability mass of Y is concentrated at b. Since in the next step this inner integral is replaced by $\frac{C}{v+1}(b-s)^{v+1}$, the rest of the proof is still valid after simply replacing all occurrences of $C/(v+1)$ by 1, and all other occurrences of v by -1; note that $Cv! = C(v+1)!/(v+1) = 1$ in this case. Thus, we see that the theorem also holds for deterministic Y. □

Alternative proof

In [dBN98], a different proof was presented for the asymptotic efficiency of this change of measure. The present proof is a bit simpler (not counting the proof of the extended central limit theorem, which will be used again in the next chapter), and yields a slightly stronger result: in [dBN98], it is only shown that $n^{\nu/2+3/4}$ is an upper bound on the asymptotic growth rate of the relative error, whereas the present proof shows this to be the exact limit.

5.2 Application: consecutive-cell-loss probability

In this section, the application is presented which originally motivated the search for an efficient change of measure for estimating the IID sum "underflow" probability (5.1). This is the problem of estimating the consecutive-cell loss (CCL) probability in an $M/G/1$ queue. We first derive an explicit expression for the CCL probability in terms of other probabilities which are simpler to estimate than the CCL probability itself. For each of these other probabilities, an asymptotically efficient importance sampling scheme is proposed, resulting in an asymptotically efficient method for the estimation of the CCL probability.

5.2.1 Model and analysis

Consider an $M/G/1/k$ queue, and define the n-CCL event as the loss of n or more consecutively arriving cells (as in Chapter 3). We are interested in the probability γ_n of one or more such n-CCL events in one busy cycle. As usual, the busy cycle is defined as the interval between two arrivals which find the system empty; these are also regeneration points. Figure 5.1 shows the buffer content during one typical busy cycle.

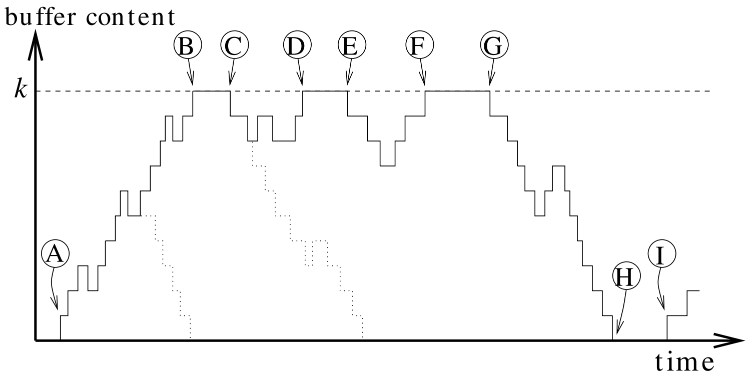


Figure 5.1: A typical regeneration cycle.

Clearly, the n-CCL event can only happen by having at least n arrivals within one full-buffer period (B–C, D–E or F–G in the figure). In a general $GI/G/1/k$ queue, the duration of full-buffer periods are neither independent, nor identically distributed. However, in an $M/G/1/k$ queue these durations are independent, and the duration of the second and later (henceforth referred to as "subsequent", as in Chapter 4) full-buffer periods are identically distributed, as can be seen easily. Therefore, the following probabilities are well-defined:

- γ: the probability of reaching full-buffer in a busy cycle (i.e., reaching level k, starting from level 0 and before reaching 0 again). In Figure 5.1, this is the probability of going from A to B.
- ϕ: the probability of reaching yet another full-buffer period in the same busy cycle (i.e., reaching level k, starting from level $k-1$ and before hitting level 0). In the figure, this corresponds to going from C to D or from E to F. The alternative is shown as going from G to H.
- p_{1n}: the probability of n or more arrivals during the first full-buffer period in a busy cycle (shown as the interval B-C in the figure).
- p_n: the probability of n or more arrivals during a subsequent full-buffer period (e.g., the intervals D-E and F-G in the figure).

Using these definitions, γ_n can be written as follows:

$$\gamma_n = \gamma p_{1n} + \frac{\gamma(1-p_{1n})\phi p_n}{1-\phi(1-p_n)} \tag{5.4}$$

The first term on the right-hand side is the probability that the n-CCL event happens in the first full-buffer period in the busy cycle; the second term is the probability that it happens in one of the subsequent full-buffer periods in the same busy cycle (using the closed form expression for the sum of a geometric series). Similarly, the four probabilities γ, ϕ, p_{1n} and p_n can also be used as a basis for calculating other quantities of interest, such as the steady-state frequency of the n-CCL event.

5.2.2 Importance sampling simulation method

Typically, the probability ϕ is of order 1, and therefore easy to estimate using standard simulation. The probability γ may be small for large k, so standard simulation may not be acceptable; but an asymptotically efficient change of measure for this estimation is well-known [PW89]. This leaves only the problem of estimating p_{1n} and p_n, which are typically also small.

Let us first consider estimating p_{1n}. This is the probability that at least n cells arrive (and are lost) during the first full-buffer period in a busy cycle. Denote the duration of such a full-buffer period as B_1; then, this probability can be expressed as follows:

$$p_{1n} = \mathbb{P}\left\{\sum_{i=1}^{n} X_i < B_1\right\}.$$

The X_i's are i.i.d. inter-arrival times whose distribution is known; for the $M/G/1$ model considered in this chapter, this distribution is exponential. However, the distribution of B_1 is not known. The only way to obtain samples of B_1 is by starting a simulation at the regeneration point corresponding to arrival to an empty system. But if we start the simulation from that point, we could just as well use the first part of the simulation (up to full-buffer) for estimating γ. So, we propose the following procedure: perform a number of simulation runs, each starting with an empty system. In each run, simulate until either the empty-buffer or the full-buffer state is reached; this part of the simulation provides observations to estimate γ. If full buffer is reached, the simulation run is continued until either the end of that full-buffer period or the n-CCL event, whichever occurs first. Simulation runs up to this point provide observations for estimating the product γp_{1n}, which is, in fact, what we need to evaluate (5.4). However, a separate estimate of p_{1n} could be obtained by taking as observations only the last part (i.e., the full buffer part) of those simulation runs which did reach full-buffer.

Estimating p_n is an almost identical problem: replace B_1 by the duration of the subsequent full-buffer periods, and use the end of full-buffer periods as starting points; then the very same procedure as described above yields estimates of ϕ and ϕp_n.

Finally, we only need to choose an appropriate change of measure to be used in the importance sampling simulations. A good (or optimal) change of measure is known for estimating each of the probabilities separately: for estimating γ, the method given in [PW89] can be used; for estimating ϕ, standard simulation can be used; and for estimating p_n and p_{1n}, an asymptotically efficient change of measure has been presented in Section 5.1, for cases where the full-buffer duration is bounded, e.g., the $M/D/1$ queue. For cases where the full-buffer duration is not bounded, no provably efficient simulation method has yet been found. (However, some alternative methods are available; see Sections 3.1 and 4.7.3.)

5.2.3 Example: $M/D/1/k$ system

Consider the estimation of the n-CCL probabilitiy in an $M/D/1/k$ queue, with arrival rate $\lambda = 0.8$ and deterministic service time $d = 1$. As described in Section 5.2.2, two separate simulations are performed. In the first simulation (to estimate γ and γp_{1n}) we use exponential tilting according to [PW89] up to full-buffer, and exponential tilting according to (5.2) during the full-buffer period. In the other simulation (to estimate ϕ and ϕp_n) we use standard simulation up to

n	γp_{1n}	rel.error	ϕp_n	rel.error
1	$2.100 \cdot 10^{-3}$	0.15 %	$2.487 \cdot 10^{-1}$	0.078 %
2	$3.653 \cdot 10^{-3}$	0.20 %	$5.798 \cdot 10^{-2}$	0.099 %
4	$6.598 \cdot 10^{-6}$	0.30 %	$1.615 \cdot 10^{-3}$	0.14 %
8	$4.594 \cdot 10^{-10}$	0.54 %	$1.957 \cdot 10^{-7}$	0.22 %
16	$3.861 \cdot 10^{-20}$	1.1 %	$3.106 \cdot 10^{-17}$	0.36 %
32	$2.278 \cdot 10^{-44}$	2.5 %	$3.377 \cdot 10^{-41}$	0.61 %
64	$9.370 \cdot 10^{-102}$	5.9 %	$2.785 \cdot 10^{-98}$	1.0 %

n	γ_n (sim.)	rel.error	γ_n (exact)
1	$6.353 \cdot 10^{-3}$	0.3 %	$6.352 \cdot 10^{-3}$
2	$2.465 \cdot 10^{-3}$	0.3 %	$2.464 \cdot 10^{-3}$
4	$8.401 \cdot 10^{-5}$	0.3 %	$8.398 \cdot 10^{-5}$
8	$9.901 \cdot 10^{-9}$	0.4 %	$9.91 \cdot 10^{-9}$
16	$1.538 \cdot 10^{-18}$	0.5 %	$1.542 \cdot 10^{-18}$
32	$1.649 \cdot 10^{-42}$	0.8 %	$1.683 \cdot 10^{-42}$
64	$1.360 \cdot 10^{-99}$	1.2 %	$1.363 \cdot 10^{-99}$

Table 5.1: Simulation results for an $M/D/1/10$ queue, with $\lambda = 0.8$ and $d = 1$.

full-buffer, and then exponential tilting according to (5.2) during the full-buffer period. To apply (5.2), we need values for q and b. Since the inter-arrival time distribution is exponential, $q = 1$. And since the service time is deterministic and equal to d, the remaining service time is a random variable assuming values between 0 and d, so $b = d = 1$. It follows that the tilting parameter during (both the first and subsequent) full-buffer periods is simply given by $\theta = n$.

Table 5.1 shows simulation results at buffer size $k = 10$, for several values of n. Each estimate is based on $4 \cdot 10^6$ observations; the accuracy of these estimates is shown by listing the estimate's relative error: the standard deviation as a percentage of the mean. Along with the results in the table, the simulations have also yielded estimates for γ and ϕ; typical values and their relative errors are: $\gamma = 9.837 \cdot 10^{-3} \pm 0.068\%$ and $\phi = 0.7961 \pm 0.025\%$. From these simulation results, γ_n and an estimate of its relative error can be calculated using (5.4). The resulting estimates of γ_n and their relative errors[2] are shown in the lower part of Table 5.1. For comparison, the exact values (obtained numerically, as described in Chapter 3) are also shown in that table; clearly, the exact results and the simulation results agree well.

[2]These are worst case relative error estimates, since we have assumed that each quantity in the r.h.s. of (5.4) deviates in the direction which increases the total error in γ_n. Then calculating the resulting relative error in γ_n from the known errors (bounds) in the quantities in the r.h.s. is straightforward.

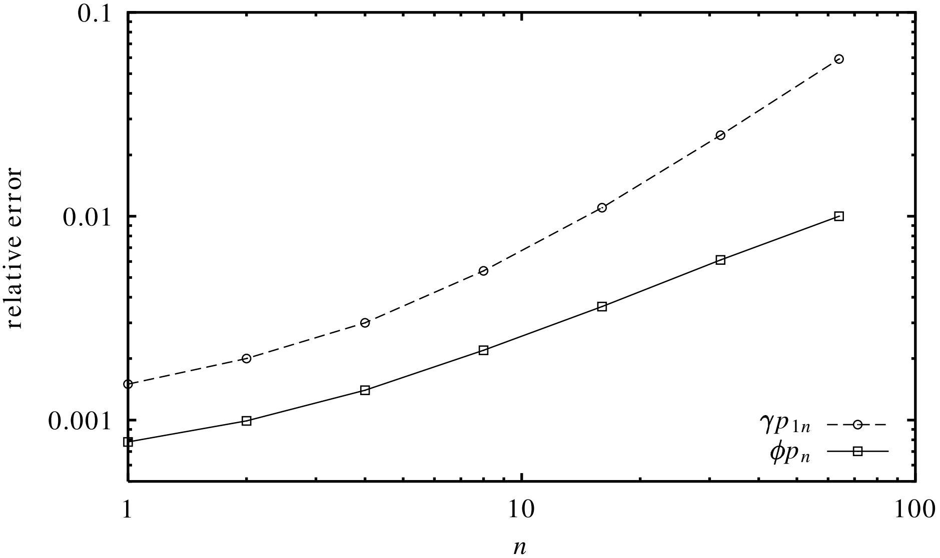


Figure 5.2: Relative error as a function of n.

We now proceed to compare the asymptotic properties of the relative error estimates with the claims of Theorem 5.1. In the estimation of p_n (involving subsequent full-buffer periods), the remaining service time B upon reaching full-buffer would asymptotically (for $k \to \infty$) have a uniform distribution on $[0, d]$ (see Chapter 4). As a consequence, the actual distribution of B (for finite k) is approximately uniform on $[0, d]$, and hence $v = 0$. Therefore, according to Theorem 5.1, we should expect the relative error to (asymptotically) grow proportionally to $n^{3/4}$. As the solid line in Figure 5.2 shows, the relative error from the ϕp_n simulations indeed grow proportionally to $n^{3/4}$ for large n.

For the estimation of p_{1n} (involving the first full-buffer period), things are a bit more complicated. Recall that the samples of the remaining service time here are obtained from an importance sampling simulation with tilting according to [PW89]: this tilting basically sets the arrival rate to 1.23084, thus making the queue unstable. According to Chapter 4, in a queue with Markov arrivals at rate 1.23084 and deterministic service times equal to 1, the remaining service time at the first full-buffer hit asymptotically has a density proportional to $\left(e^{-0.35004(1-x)} - 1\right)$. Thus $v = 1$, so according to Theorem 5.1, we would expect the relative error in p_{1n} to (asymptotically) grow proportional to $n^{5/4}$. Actually, this expectation is not completely justified since the change of measure during the first part of the simulation (up to full buffer) has an influence on the likelihood ratios, while the theorem does not allow for such an influence. However, the experimental results (dashed line) plotted in Figure 5.2 confirm the growth proportional to $n^{5/4}$, and actually, the proof of the theorem can be extended for this particular case[3].

5.3 Conclusions

In this chapter, we have proposed an asymptotically efficient importance sampling scheme for the estimation of probabilities of the form $\mathbb{P}(\sum_{i=1}^{n} X_i < Y)$, where the X_i are independent and identically distributed, and Y is some bounded random variable. We have derived expressions for the asymptotic growth rate of the relative error with n.

As an example application, the estimation of the consecutive cell loss probability in $M/D/1/k$ queues has been considered, with good results (confirming the theory).

The importance sampling simulation method described in the present chapter involves only changing the distribution of the X_i and not of Y. This restriction can have implementation advantages, especially if the distribution of Y is not explicitly known. However, this restriction also makes the method limited to cases in which Y is upper-bounded. In the next chapter, a method is presented which changes the distributions both of X_i and of Y, and works also for unbounded Y.

5.A Extension of the Central Limit Theorem to exponentially tilted random variables

The well-known Central Limit Theorem (CLT) basically states that the distribution of the sum of n i.i.d. random variables approaches a normal distribution when n goes to infinity. In this appendix, a variant of the CLT is presented and proved in which the distribution of the random variables depends on n; within one sum for a given n, the random variables are still i.i.d.. Furthermore, it is shown that not just the distribution function, but also the density converges[4] to normal.

[3]One can easily verify that the proof in Section 5.1.2 remains valid if the likelihood ratio L is multiplied by a function of (only) Y and that function does not go to zero at $y = b$. Furthermore, one can verify that the likelihood ratio at the end of a sample path with tilting according to [PW89], is indeed such a function of the remaining service time Y.

[4]The usual CLT only claims convergence of the distribution function. For convergence of the density, some additional conditions are necessary; see e.g. Theorem XV.5.2 in [Fel66].

5.A.1 A preliminary lemma

Lemma 5.1 *Consider a family of random variables X, parameterized by θ, with a density $f_X(\cdot)$ satisfying*[5]*:*

$$f_X(x) = \begin{cases} 0 & \text{for } x < 0 \\ \rho_X(\theta)\left(x^r + o(x^r)\right) e^{-\theta x} & \text{for } x \geq 0, \end{cases}$$

for some positive constant r, and $\theta \geq 1$. Here $\rho_X(\theta)$ is a normalization factor, such that $\int_0^\infty f_X(x)dx = 1$. Denote the expectation of X by μ, and its standard deviation by σ, which obviously are both functions of θ. Then the sum $S = \sum_{i=1}^n X_i$ of n independent such random variables has a density $f_S(\cdot)$ with the following asymptotic behaviour as $n \to \infty$:

$$\lim_{n\to\infty} \frac{f_S(x) - \mathfrak{n}_{n\mu,n\sigma^2}(x)}{1/\left(\sigma\sqrt{n}\right)} = 0,$$

where $\mathfrak{n}_{n\mu,n\sigma^2}(\cdot)$ denotes a normal density with mean $n\mu$ and variance $n\sigma^2$. The convergence is uniform in x, and holds even if θ (and thus μ and σ) vary with n.

Proof: One can easily show that the expectation μ of X is given by

$$\mu = \frac{r+1}{\theta} + o(\theta^{-1})$$

and its variance σ^2 by

$$\sigma^2 = \frac{r+1}{\theta^2} + o(\theta^{-2}).$$

Define a family of random variables $Y = (X - \mu)/\sigma$; clearly, $\mathbb{E}Y = 0$ and $\mathbb{E}Y^2 = 1$. The density $f_Y(\cdot)$ of Y is given by

$$f_Y(y) = \begin{cases} 0 & \text{for } y < -\mu/\sigma \\ \rho'_Y(\theta)\,(y\sigma + \mu)^r \left(1 + w_1(y\sigma + \mu)\right) e^{-\theta(y\sigma+\mu)} & \text{for } y \geq -\mu/\sigma, \end{cases}$$

where $w_1(\cdot)$ is some function such that $\lim_{x\to 0} w_1(x) = 0$, and where $\rho'_Y(\theta)$ is a normalizing factor such that $\int f_Y(y)dy = 1$. Define $\beta = 1/\theta$. Consequently, $\mu = \beta(r+1) + o(\beta)$ and $\sigma^2 = \beta^2(r+1) + o(\beta^2)$. From this, it follows that $\mu/\sigma = \sqrt{r+1} + w_2(\beta)$, where $w_2(\beta)$ is a function such that $\lim_{\beta\to 0} w_2(\beta) = 0$. Then

$$f_Y(y) = \begin{cases} 0 & \text{for } y < -\sqrt{r+1} - w_2(\beta) \\ \rho_Y(\beta)\left(y + \sqrt{r+1} + w_2(\beta)\right)^r \left(1 + w_1(y\sigma + \mu)\right) e^{-y\sqrt{r+1}} & \\ & \text{for } y \geq -\sqrt{r+1} - w_2(\beta), \end{cases}$$

where $\rho_Y(\beta)$ is again a normalization factor. Note that the original interval $[1, \infty[$ for θ corresponds to the interval $]0, 1]$ for β. However, the above expression for

[5] The small order symbol $o(x)$ is defined here by $\lim_{x\to 0} \frac{o(x)}{x} = 0$.

$f_Y(y)$ does not have singularities at $\beta = 0$, so the range of β can be extended to the closed interval $[0, 1]$. For every β the density $f_Y(y)$ satisfies the conditions of Theorem XV.5.2 in [Fel66], according to which the density of $\sum_{i=1}^{n} Y_i/\sqrt{n}$ converges uniformly to the standard normal density function $e^{-y^2/2}/\sqrt{2\pi}$. Since this holds for any constant β in the closed interval $[0, 1]$, it also holds if, with increasing n, β varies within this interval, as can easily be verified[6]. Since

$$\sum_{i=1}^{n} \frac{Y_i}{\sqrt{n}} = \frac{S - n\mu}{\sigma\sqrt{n}},$$

the density of $(S - n\mu)/\sigma\sqrt{n}$ also converges uniformly to the standard normal density. From this, the convergence claimed in the lemma follows immediately.

5.A.2 The extended central limit theorem

Theorem 5.2 *Consider random variables X_i with a distribution function $F_X(\cdot)$ satisfying $F_X(0) = 0$ (i.e., X is a positive random variable) and*

$$\frac{dF_X(x)}{dx} = \rho_X \left(x^r + o(x^r)\right) \quad \text{for } 0 \leq x \leq a,$$

for some positive constants ρ_X, r and a. No restrictions on the behaviour of the distribution of X in the interval $[a, \infty[$ are needed.

Next, consider the random variables Y_i, obtained from X_i by negatively exponentially tilting the distribution:

$$dF_Y(y) = \rho_Y(\theta)e^{-\theta y}dF_X(y)$$

where $\theta > 0$ is the tilting parameter and ρ_Y normalizes the function such that $F_Y(\infty) = 1$. The tilting parameter θ must be such, that as $n \to \infty$, it increases at least polynomially with n; i.e., $\exists p > 0 : \lim_{n\to\infty} \theta/n^p > 0$.

Then the random variable $S = \sum_{i=1}^{n} Y_i$ has a density on the interval $[0, na]$, and this density converges to a normal density with mean $\frac{n(r+1)}{\theta}$ and variance $\frac{n(r+1)}{\theta^2}$ for $n \to \infty$, as follows:

$$\lim_{n\to\infty} \frac{f_S(y) - \mathfrak{n}_{n(r+1)/\theta, n(r+1)/\theta^2}(y)}{\theta/\sqrt{n(r+1)}} = 0 \tag{5.5}$$

uniform in y.

[6]From Theorem XV.5.2 in [Fel66], we know that for any $\epsilon > 0$, and for any $\beta \in [0, 1]$, an $N_0(\beta, \epsilon)$ can be given such that for all $n > N_0(\beta, \epsilon)$ the maximum difference between the true density of $\sum_{i=1}^{n} Y_i/\sqrt{n}$ and the standard normal density is less than ϵ. Now define $N(\epsilon) = \max_{\beta\in[0,1]} N_0(\beta, \epsilon)$. Clearly then, for any $\epsilon > 0$, the difference between the true density and the standard normal density is less than ϵ if $n > N(\epsilon)$, independent of β.

Proof: Start by writing the distribution function of Y in the following way:

$$F_Y(y) = F_Y(a)G(y) + H(y),$$

where

$$G(y) = \begin{cases} 0 & \text{for } y < 0 \\ \frac{F_Y(y)}{F_Y(a)} & \text{for } 0 \le y \le a \\ 1 & \text{for } y > a \end{cases}$$

and

$$H(y) = \begin{cases} 0 & \text{for } y \le a \\ F_Y(y) - F_Y(a) & \text{for } y > a. \end{cases}$$

Note that the maximum of $H(y)$ is $1 - F_Y(a)$, which vanishes as $\theta \to \infty$:

$$\begin{aligned} 1 - F_Y(a) &= \frac{\int_a^\infty e^{-\theta y} dF_X(y)}{\int_0^\infty e^{-\theta y} dF_X(y)} \le \frac{e^{-\theta a} \int_a^\infty dF_X(y)}{\int_0^a e^{-\theta y} dF_X(y)} \\ &\le \frac{e^{-\theta a}}{\rho_X \int_0^a e^{-\theta x}\left(x^r + o(x^r)\right) dx} \propto e^{-\theta a}\theta^{r+1}. \end{aligned} \tag{5.6}$$

In order to find the probability distribution of S, we need to calculate the n-fold convolution of $F_Y(\cdot)$:

$$F_S(y) = F_Y^{n*}(y) = \underbrace{F_Y^n(a)G^{n*}(y)}_{\text{(a)}} + \underbrace{H^{n*}(y)}_{\text{(b)}} + \underbrace{\sum_{i=1}^{n-1} \binom{n}{i} F_Y(a)^i G^{i*}(y) * H^{(n-i)*}(y)}_{\text{(c)}}.$$

The terms (a), (b) and (c) in the above can be studied separately:
(a) According to Lemma 5.A.1, the term (a) converges exactly as claimed (for the whole distribution) in (5.5). Thus, it remains to be shown that the other terms vanish.
(b) Since $H(y) = 0$ for $y \le a$, the term (b) is zero for all $y \le na$, so it cannot disturb the density there. Furthermore, for $y > na$ its influence vanishes as n (and thus θ) go to infinity.
(c) This term is a sum of convolutions, each of which contains at least one factor $G(y)$. Since $G(y)$ is differentiable (i.e., it has a density), convolutions of it are also differentiable. So the term (c) has a density, and this density can be upper

bounded as follows[7]:

$$
\begin{aligned}
\frac{d(\mathrm{c})}{dy} &= \frac{dG(y)}{dy} * \sum_{i=1}^{n-1} \binom{n}{i} F_Y(a)^i G^{(i-1)*}(y) * H^{(n-i)*}(y) \\
&\le \frac{dG(y)}{dy} * H(y) * \sum_{j=0}^{n-2} \binom{n}{j+1} F_Y(a)^j G^{j*}(y) * H^{(n-2-j)*}(y) \\
&\le \frac{dG(y)}{dy} * H(y) * n^2 \sum_{j=0}^{n-2} \binom{n-2}{j} F_Y(a)^j G^{j*}(y) * H^{(n-2-j)*}(y) \\
&= \frac{dG(y)}{dy} * H(y) * n^2 F_Y^{(n-2)*}(y) \\
&\le n^2 \frac{dG(y)}{dy} * H(y).
\end{aligned}
$$

An upper bound for $dG(y)/dy$ can easily be derived: ignoring the $o(x^r)$ term, the maximum occurs at $y = r/\theta$ and is equal to $e^{-r} r^r \theta / r! = C\theta$, for some positive constant C. Using this, and the upper bound for $H(y)$ given by (5.6), we arrive at the following upper bound for the density contribution of term (c):

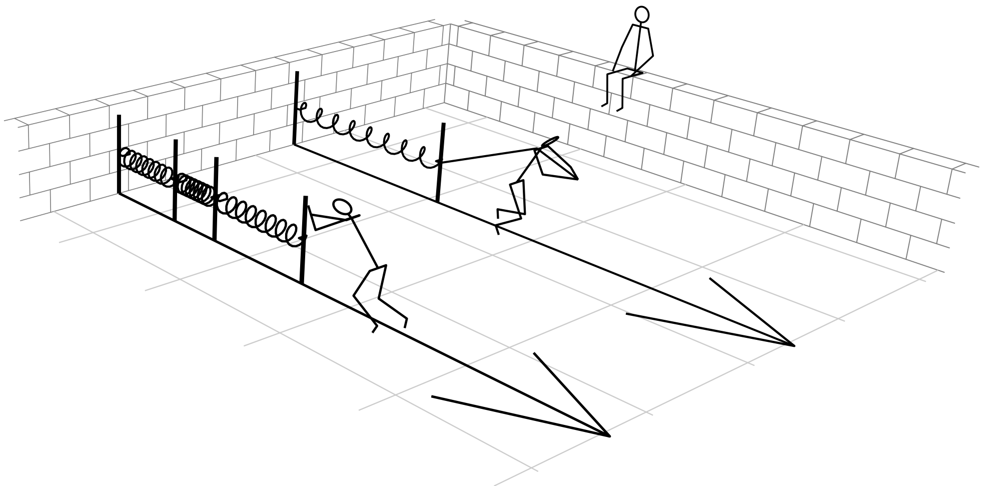
$$
\frac{d(\mathrm{c})}{dy} \le n^2 C\theta e^{-\theta a} \theta^{r+1} = n^2 C e^{-\theta a} \theta^{r+2}.
$$

Since θ increases at least polynomially with n, we find

$$
\lim_{n\to\infty} \frac{\frac{d(\mathrm{c})}{dy}}{\theta/\sqrt{n}} \le \lim_{n\to\infty} \frac{n^2 C e^{-\theta a} \theta^{r+2}}{\theta/\sqrt{n}} = 0.
$$

Therefore, the contribution of the term (c) to the density of S is negligible in the sense that it does not disturb the convergence of (5.5).

[7] Note that from $(f * g)(y) = \int f(x)\, g(y-x) dx$, it follows that $(f * g)'(y) = \int f(x)\, g'(y-x) dx = (f * g')(y)$.



Chapter 6

Rare events involving IID sums: unbounded threshold

Like in the previous chapter, in this chapter techniques for estimating probabilities of the form

$$\gamma = \mathbb{P}\left(\sum_{i=1}^{n} X_i < Y\right) \tag{6.1}$$

are studied, where X_i are positive i.i.d. random variables, and the "threshold" Y is another independent positive random variable. As explained at the beginning of the previous chapter, such probabilities occur in various performance and reliability models. We again focus on efficient estimation of these (typically very small) probabilities using importance sampling simulation, and also derive an analytical approximation.

The change of measure used for the importance sampling procedure in the previous chapter was quite restricted: the distribution of Y was left unchanged, and an exponential tilting was applied to the X_i. In the following we again use exponential tilting, applied not only to X_i, but also to Y. This has several consequences:

- The relative error of the resulting estimators increases slower with increasing n.

- The previous chapter's restriction to models in which Y is upper bounded is relaxed. However, the method cannot be applied in situations where the distribution of Y has a sub-exponential (heavy) tail.

- In problems where the samples of Y are obtained empirically (i.e., from a preceding simulation, such as in the example of Section 5.2), it is generally not possible to change the distribution, so this chapter's method cannot be applied to those problems.

In Section 6.1, we discuss the exponential change of measure to be used for the estimation of this probability. This change of measure yields asymptotically efficient estimators; it is formally shown that the relative error for a fixed number of replications asymptotically increases no faster than $\sqrt{n}$. In Section 6.2, the same change of measure is used to obtain a tight analytical approximation for this probability for large n. Finally, in Section 6.3, we give three examples from different application areas, and we use these to examine the validity and accuracy of both the importance sampling simulation scheme and the analytical approximation.

6.1 Importance sampling simulation

In this section, the importance sampling simulation is discussed. In Section 6.1.1, the tilting method is described, and in Section 6.1.2, its asymptotic efficiency is considered.

6.1.1 The tilting method

To estimate γ using importance sampling simulation, we propose the following exponential change of measure:

- The distribution of X_i is tilted exponentially[1]: $f_X^*(x) = f_X(x)e^{-\theta x}/M_X(-\theta)$, where θ is the tilting parameter.
- The distribution of Y is also tilted exponentially, equally strong, but in the opposite direction: $f_Y^*(y) = f_Y(y)e^{\theta y}/M_Y(\theta)$.
- The tilting parameter θ is chosen such that:

$$n\mathbb{E}^* X_i = \mathbb{E}^* Y. \tag{6.2}$$

  In words: the expectation of the sum of the tilted X_i is equal to the expectation of the tilted Y. Note that this means that the tilting parameter θ depends on n.

Together, these rules define the tilting which will be used not only for the importance sampling simulation, but also for the analytical approximation in Section 6.2.

[1] We use the following (common) notational conventions: $F_X(\cdot)$ is the probability distribution function of X; $f_X(\cdot)$ is the probability density function of X; $M_X(s) = \mathbb{E}e^{sX}$ is the moment generating function of the density of X; a star ($*$) is used to denote tilted distributions, expectations, etc.

The importance sampling simulation procedure is as follows. Perform a large number, say N, of "replications", where each replication consists of sampling once from the tilted Y distribution (yielding y), and n times[2] from the tilted X_i distribution (yielding $x_1, \ldots, x_n$). For each replication, the indicator $I_{\Sigma x_i < y}$ is determined, along with the likelihood ratio

$$L = M_X^n(-\theta) M_Y(\theta) e^{-\theta(Y - S_n)}, \tag{6.3}$$

where $S_n = \sum_{i=1}^n X_i$. The estimate $\hat{\gamma} = \sum_{k=1}^N L_k I_k / N$ is unbiased, and its variance is estimated by $\hat{\sigma}_\gamma^2 = \frac{1}{N(N-1)} \sum_{k=1}^N (L_k I_k - \hat{\gamma})^2$.

Clearly, for the rare event $\sum X_i < Y$ to happen, the X_i's should be small and Y should be large. In other words, this probability is mostly determined by the behaviour of X_i near zero and by the tail behaviour of Y. For some typical tails of Y, we show analytically that the proposed tilting results in asymptotically efficient simulation, and we obtain an upper bound on the asymptotic growth of the relative error in $\hat{\gamma}$ with n (for a fixed number of simulation replications N). These cases are:

- Y has an exponential tail (e.g., phase-type distributions). In this case, the relative error asymptotically increases proportional to $n^{1/2}$.

- Y has a super-exponential tail, satisfying a few technical conditions given at the beginning of Section 6.1.2. This includes distributions such as positive normal and Weibull. In this case, the relative error asymptotically grows proportional to $n^{1/4}$.

- Y is upper bounded (e.g., uniform and deterministic distributions). This is a special case of the previous one, so again the relative error asymptotically grows proportional to $n^{1/4}$. Note that in this case the method from Chapter 5 (which does not tilt Y) could also be applied; however, this would result in a relative error that grows faster than $n^{1/4}$ (except for the case of deterministic Y, where both methods are equivalent).

As noted before, the method is not applicable to cases where Y has a sub-exponential tail (heavy tail): such distributions cannot be tilted exponentially to favor larger Y.

[2] In practice, one can save some computational effort by stopping after $k < n$ samples, as soon as $\sum_{i=1}^k x_i > y$.

6.1.2 Asymptotic efficiency

As already noted in Section 5.1.2, the relative error of an importance sampling simulation is given by

$$RE = \sqrt{\frac{\mathbb{E}^* L^2 I}{(\mathbb{E}^* L I)^2} - 1} \leq \sqrt{\frac{\mathbb{E}^* L^2 I}{(\mathbb{E}^* L I)^2}},$$

where L is the likelihood ratio as given by (6.3), and I is the indicator function $I_{S_n < Y}$. An upper bound for $\mathbb{E}^*(L^2 I)$ can be derived as follows:

$$\mathbb{E}^*(L^2 I) = \mathbb{E}(LI) = M_X^n(-\theta)\, M_Y(\theta)\, \mathbb{E}\left(e^{-\theta(Y - S_n)}\, I_{Y > S_n}\right) \leq M_X^n(-\theta)\, M_Y(\theta)\, \mathbb{E}I.$$

Therefore, an upper bound α for the square of the relative error is given by

$$\alpha = \frac{M_X^n(-\theta)\, M_Y(\theta)}{\mathbb{E}I}.$$

For showing asymptotic efficiency, it is sufficient to show that α increases at most polynomially with n.

In the rest of this chapter, we will repeatedly discuss the mean and variance of exponentially tilted random variables. Let Z be a random variable and ϑ the tilting parameter; the tilted density is $f_Z^*(z) \propto e^{\vartheta z} f_Z(z)$. Then the following convenient expressions can easily be derived:

$$\mathbb{E}^* Z = \frac{M_Z'(\vartheta)}{M_Z(\vartheta)} = \frac{d}{d\vartheta} \ln M_Z(\vartheta) \tag{6.4}$$

$$\mathrm{Var}^* Z = \frac{d^2}{d\vartheta^2} \ln M_Z(\vartheta) = \frac{d}{d\vartheta} \mathbb{E}^* Z.$$

Case 1: Y has an exponential tail

For simplicity, assume that Y is exponentially distributed with parameter ν. Since only the tail behaviour of Y is important, this should not really restrict the validity of the result. In this case, $\mathbb{E}I$ can be calculated explicitly:

$$\begin{aligned}\mathbb{E}I = \mathbb{P}(S_n < Y) &= \int_0^\infty \mathbb{P}(S_n < y) dF_Y(y) = \int_0^\infty \nu e^{-\nu y} \int_0^y dF_{S_n}(z)\, dy \\ &= \int_0^\infty \int_z^\infty \nu e^{-\nu y} dy\, dF_{S_n}(z) = \int_0^\infty e^{-\nu z} dF_{S_n}(z) = M_X^n(-\nu).\end{aligned}$$

Consequently:

$$\alpha = \frac{\nu}{\nu - \theta} \left(\frac{M_X(-\theta)}{M_X(-\nu)}\right)^n.$$

From (6.2):

$$n \frac{M_X'(-\theta)}{M_X(-\theta)} = \frac{1}{\nu - \theta}$$

or

$$(\nu-\theta)M_X'(-\theta)=\frac{M_X(-\theta)}{n}, \tag{6.5}$$

so as n approaches infinity, θ approaches ν. Write $M_X(-\nu)$ as a Taylor series[3]:

$$M_X(-\nu)=M_X(-\theta)-(\nu-\theta)M_X'(-\theta)+\mathcal{O}\left((\nu-\theta)^2\right)=M_X(-\theta)\left(1-\frac{1}{n}\right)+\mathcal{O}\left((\nu-\theta)^2\right)$$

This allows us to write α for large n as

$$\alpha\approx\nu\frac{n\,M_X'(-\theta)}{M_X(-\theta)}\left(1-\frac{1}{n}\right)^{-n}\approx\frac{\nu\,M_X'(-\theta)}{M_X(-\theta)\,e^{-1}}\,n$$

Thus, α increases approximately linearly in n; therefore, the simulation scheme is asymptotically efficient, with a relative error increasing at most proportionally to $n^{1/2}$.

Case 2: Y has a super-exponential tail

In this section, we make the following additional assumptions:

- X satisfies the conditions of the extended central limit theorem, Theorem 5.2 in the appendix of Chapter 5. These conditions basically require that X has a finite non-zero density near 0.
- Y is such that

$$\frac{\mathrm{Var}^*\,Y}{\mathbb{E}^*Y}=\mathcal{O}(\theta^{-1}); \tag{6.6}$$

  a sufficient[4] condition for this is

$$\lim_{\theta\to\infty}\frac{\ln\mathbb{E}^*Y}{\ln\theta}=C,$$

  for some positive constant C. Any system where $\mathbb{E}^*Y$ grows polynomially with θ, for large θ, satisfies this condition; this includes many typical super-exponential distributions.

First, write the probability of interest $\mathbb{E}I$ as follows:

$$\begin{aligned}\mathbb{E}I&=\int_0^\infty\int_0^\infty I_{z<y}\,dF_{S_n}(z)\,dF_Y(y)=\int_0^\infty\int_0^\infty L\,I_{z<y}\,dF_Y^*(y)\,dF_{S_n}^*(z)\\&=M_X^n(-\theta)M_Y(\theta)\int_0^\infty\int_0^\infty e^{-\theta(y-z)}I_{z<y}\,dF_Y^*(y)\,dF_{S_n}^*(z).\end{aligned}$$

[3] The big order symbol $\mathcal{O}(x)$ is defined by $\lim_{x\to0}\frac{\mathcal{O}(x)}{x}=C$ with $C\in\mathbb{R}$.

[4] Applying L'Hospital's Rule gives $\lim_{\theta\to\infty}\frac{\mathrm{Var}^*\,Y/\mathbb{E}^*Y}{1/\theta}=C$, which is equivalent to (6.6).

Then

$$\alpha = \frac{M_X^n(-\theta)M_Y(\theta))}{\mathbb{E}I} = \frac{1}{\int_0^\infty \int_z^\infty e^{-\theta(y-z)} dF_Y^*(y)\, dF_{S_n}^*(z)}.$$

We need to prove that this α increases less than exponentially fast with n. To do so, start by rewriting $1/\alpha$ as follows (for brevity we write S instead of S_n):

$$\frac{1}{\alpha} = \int_0^\infty \int_z^\infty e^{-\theta(y-z)}\, dF_Y^*(y)\, dF_S^*(z) = \int_0^\infty \int_0^y e^{-\theta(y-z)}\, dF_S^*(z)\, dF_Y^*(y).$$

Restrict both ranges of integration to obtain a lower bound on $1/\alpha$:

$$\frac{1}{\alpha} \geq \int_{m-a\sqrt{n}/\theta}^{m+a\sqrt{n}/\theta} \int_{y-(1/\theta)}^{y} e^{-\theta(y-z)} dF_S^*(z)\, dF_Y^*(y),$$

where $m = \mathbb{E}^*Y = \mathbb{E}^*S$, and a is a positive constant which will be chosen later. In the above range for z, we have $e^{-\theta(y-z)} \geq e^{-1}$, so

$$\frac{1}{\alpha} \geq \int_{m-a\sqrt{n}/\theta}^{m+a\sqrt{n}/\theta} \int_{y-(1/\theta)}^{y} e^{-1} dF_S^*(z)\, dF_Y^*(y).$$

The inner integral can be lower bounded by e^{-1}/θ times the minimum of $dF_S^*(z)/dz = f_S^*(z)$. Taking into account the range of y (as given by the outer integral), it is clear that z in the inner integral is restricted to the interval $[m-(a\sqrt{n}+1)/\theta, m+a\sqrt{n}/\theta]$. After bounding the inner integral this way independently of y, the outer integral can be written as a simple probability. We find:

$$\frac{1}{\alpha} \geq \mathbb{P}^*\left(|Y-m| \leq \frac{a\sqrt{n}}{\theta}\right) \cdot \frac{e^{-1}}{\theta} \cdot \min_{z\in[m-(a\sqrt{n}+1)/\theta, m+a\sqrt{n}/\theta]} f_S^*(z).$$

Finally, rewrite the first factor using Chebyshev's inequality:

$$\frac{1}{\alpha} \geq \underbrace{\left(1 - \frac{\theta^2 \operatorname{Var}^* Y}{na^2}\right)}_{\text{(I)}} \cdot \frac{e^{-1}}{\theta} \cdot \underbrace{\min_{z\in[m-(a\sqrt{n}+1)/\theta, m+a\sqrt{n}/\theta]} f_S^*(z)}_{\text{(II)}}. \tag{6.7}$$

One can easily verify that asymptotically $\mathbb{E}^*X \approx (r+1)/\theta$, where r is a constant depending on the distribution of X as given in the conditions of the extended central limit theorem (Theorem 5.2); combining this with (6.2) and (6.6) yields

$$\frac{\theta^2 \operatorname{Var}^* Y}{n} = \frac{\theta^2 \operatorname{Var}^* Y \cdot \mathbb{E}^*X}{\mathbb{E}^*Y} \approx \frac{\theta^2 \operatorname{Var}^* Y \cdot (r+1)/\theta}{\mathbb{E}^*Y} = \theta\, \mathcal{O}(\theta^{-1}) = \mathcal{O}(1). \tag{6.8}$$

So a can trivially be chosen such that

$$1 - \frac{\theta^2 \operatorname{Var}^* Y}{n} \frac{1}{a^2} \geq \frac{1}{2}, \tag{6.9}$$

thus providing a lower bound for the factor (I) in (6.7). Note that instead of 1/2, any constant between 0 and 1 could have been used in the above inequality, without affecting the final result.

The factor (II) in (6.7) will be estimated using Theorem 5.2, but first we need to show that the conditions for that theorem are satisfied. Write $m_Y(\theta) = \mathbb{E}^*Y$ and $m_X(\theta) = \mathbb{E}^*X$. The tilting (6.2) prescribes that $m_Y(\theta) = nm_X(\theta)$. Differentiate this to find

$$m'_Y(\theta) = \frac{dn}{d\theta} m_X(\theta) + nm'_X(\theta),$$

from which it follows that

$$\frac{1}{n}\frac{dn}{d\theta} = \frac{m'_Y(\theta)}{nm_X(\theta)} - \frac{m'_X(\theta)}{m_X(\theta)} = \frac{m'_Y(\theta)}{m_Y(\theta)} - \frac{m'_X(\theta)}{m_X(\theta)}.$$

Now recall that $\mathrm{Var}^*\, Y = d\mathbb{E}^*Y/d\theta = m'_Y(\theta)$, and that $m_X(\theta) = (r+1)/\theta + \mathcal{O}(\theta^{-1})$, and use (6.6) to find:

$$\frac{1}{n}\frac{dn}{d\theta} = \mathcal{O}(\theta^{-1}) + \theta^{-1} \le C_1\theta^{-1},$$

for some constant C_1. By integrating we get

$$\ln n \le C_2 + C_1 \ln\theta \quad \Rightarrow \quad n \le e^{C_2}\theta^{C_1}$$

for some constant C_2. So n increases at most polynomially with θ, thus satisfying the last condition of Theorem 5.2. Hence, the density $f_S^*(\cdot)$ will asymptotically approach a normal density with a standard deviation of $\sqrt{n(r+1)}/\theta$. Consequently:

$$\min_{z\in[\mu-(a\sqrt{n}+1)/\theta,\mu+a\sqrt{n}/\theta]} f_S^*(z) \ge \frac{1}{\sqrt{2\pi}\sqrt{n(r+1)}/\theta} e^{-\frac{(a\sqrt{n}+1)^2}{2n(r+1)}} \approx \frac{C\theta}{\sqrt{n}},$$

for some positive constant C and large n.

Substituting the above and (6.9) into (6.7), we have:

$$\alpha \le \frac{1}{\frac{1}{2}\cdot\frac{e^{-1}}{\theta}\cdot\frac{C\theta}{\sqrt{n}}} = \frac{2}{Ce^{-1}}\sqrt{n}.$$

This completes the proof of the asymptotic efficiency, and implies that the relative error will asymptotically grow no faster than $n^{1/4}$.

Case 3: Y is upper bounded

This is a special case of Case 2, so the same asymptotic efficiency proof holds. □

6.2 Analytical approximation

Start by writing γ as follows

$$\gamma = \int_0^\infty (1 - F_Y(y)) dF_{\Sigma X_i} = \int_0^\infty (1 - F_Y(y)) L \, dF_{\Sigma X_i^*},$$

where X_i^* are tilted as described above, and L denotes the corresponding likelihood ratio. Note that Y is not tilted here. (This is basically the "g-method" of [Sri98b].) By replacing the distribution of X_i^* by a Gaussian distribution, one obtains the following approximation [dBNS00]:

$$\tilde{\gamma} = \frac{e^{n\mu(-\theta)}}{\sqrt{2\pi n \mu''(-\theta)}} \int_0^\infty (1 - F_Y(y)) e^{\theta y} e^{-(y - n\mu'(-\theta))^2 / 2n\mu''(-\theta)} dy. \tag{6.10}$$

Here, θ is the tilting parameter as obtained in (6.2), and $\mu(s) = \ln M_X(s)$, the log moment generating function of the density of X_i. Note that the derivatives $\mu'(-\theta)$ and $\mu''(-\theta)$ are the mean and the variance of the tilted distribution of X_i, respectively.

In Appendix 6.A, we present a theorem with its proof regarding the validity of the above approximation in the case where the distribution of Y has an exponential or a superexponential tail: the approximation asymptotically (as $n \to \infty$) converges to the true probability γ in the sense that

$$\lim_{n \to \infty} \frac{\tilde{\gamma}}{\gamma} = 1.$$

6.3 Application examples

In this section, we provide some application examples involving the estimation of probabilities of the form (6.1). For each example, we give a table with simulation estimates of the probability itself and the relative errors (standard deviation divided by the mean) of these estimates after 10^5 replications, as well as the estimate of the probability as obtained from the approximation (6.10) in Section 6.2. Whenever possible, the true value (obtained numerically) of the probability being estimated is also included, for the purpose of validation. In order to check the asymptotic efficiency of the simulation using the proposed change of measure, graphs showing the relative error (RE) as a function of n are presented. For comparison, each graph contains a dotted line with a slope (1/2 or 1/4, see Section 6.1) corresponding to the theoretical asymptotic growth of the relative error. Note that in most examples we increase n so far, that the probabilities become extremely low. These extremely low values are not really of practical interest, but we provide them to show that the asymptotic behaviour of the relative error agrees with the theoretical results established in this chapter.

6.3.1 Reliability example

Consider a critical system component which may fail, and for which n (good and non-operational) spares are available to keep the system running while the original component is being repaired. Let the r.v. Y be the repair time of the original component, and X_i be the (operational) time until failure of the ith spare component. Assume the spares are identical, so the X_i's are i.i.d. Then the probability that all n spares are exhausted before the original component is repaired, is given by $\gamma = \mathbb{P}(\sum_{i=1}^{n} X_i < Y)$.

We assume the life times X_i to have a Weibull distribution with $\lambda = 1$ and $\beta = 1.5$, and the repair time Y to be deterministic, equal to 1. The results are shown in Table 6.1 and Figure 6.1. Note that sampling from a tilted Weibull distribution can be done efficiently as described in [Nak92]),

n	$\hat{\gamma}$ (IS simulation)	RE	$\tilde{\gamma}$ (anal. approx.)	γ (numerical)
4	$2.752 \cdot 10^{-3}$	± 0.44 %	$2.460 \cdot 10^{-3}$	$2.745 \cdot 10^{-3}$
8	$1.392 \cdot 10^{-8}$	± 0.58 %	$1.315 \cdot 10^{-8}$	$1.402 \cdot 10^{-8}$
16	$1.145 \cdot 10^{-22}$	± 0.73 %	$1.114 \cdot 10^{-22}$	$1.153 \cdot 10^{-22}$
32	$5.914 \cdot 10^{-58}$	± 0.90 %	$5.806 \cdot 10^{-58}$	$5.91 \cdot 10^{-58}$
64	$7.040 \cdot 10^{-143}$	± 1.09 %	$7.029 \cdot 10^{-143}$	$7.09 \cdot 10^{-143}$
128	$1.729 \cdot 10^{-341}$	± 1.29 %	$1.685 \cdot 10^{-341}$	–
256	$2.026 \cdot 10^{-796}$	± 1.58 %	$2.05 \cdot 10^{-796}$	–

Table 6.1: Results for the reliability example.

The agreement of the probability estimates from simulation and through numerical means (evaluating the convolution integral) is evident. The graph shows that the relative error asymptotically increases proportional to $n^{1/4}$ (indicated by the dotted line), which agrees with our theoretical bound. Furthermore, the analytical approximation (6.10) for the probability also agrees with the simulation and numerical estimates.

6.3.2 Queueing example

Consider a simple queueing system with finite buffer size. When the buffer is full, the system rejects arriving customers (e.g., cells in an ATM system). An interesting performance measure is the probability that during one full-buffer interval, n or more (necessarily consecutive) arrivals are blocked. If we denote the duration of the full-buffer period by Y and assume that the inter-arrival times of the cells are given by X_i (i.i.d.), this probability is given by $\gamma = \mathbb{P}(\sum_{i=1}^{n} X_i < Y)$.

Consider an $M/H_2/1$ system, with arrival rate 0.8, and a two-stage hyperexponential service process with rates 2 and 2/3, each with probability 1/2. Ac-

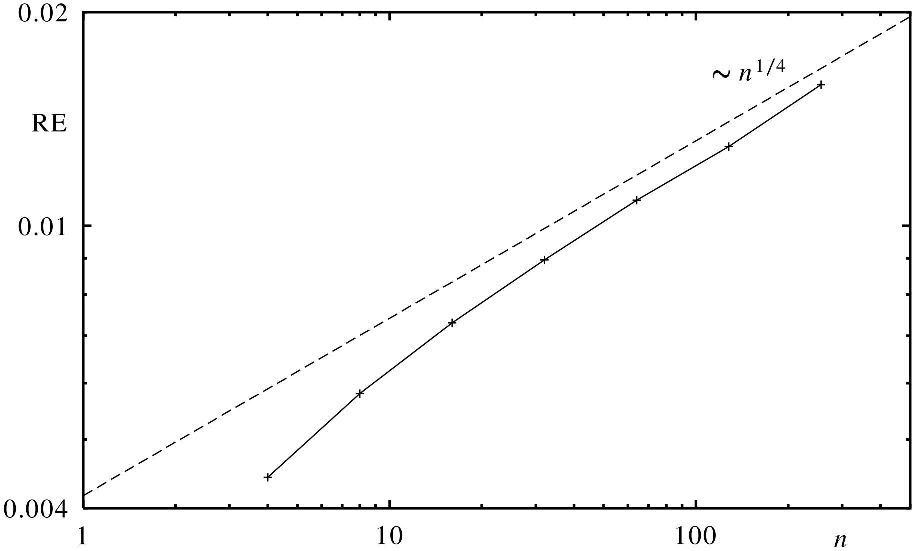


Figure 6.1: Relative error for the reliability example.

cording to Chapter 4, the asymptotic distribution (for large buffer size) of the duration of "subsequent" full-buffer periods is also hyperexponentially distributed, with rates 2 and 2/3 and probabilities 1/4 and 3/4, respectively. Because Y has an exponential tail, the theoretical asymptotic bound for the relative error is $n^{1/2}$. Results are shown in Table 6.2 and Figure 6.2. Note that due to the simple distribution of Y and X_i, an exact analytical expression for γ can easily be derived and is used to obtain the numerical results in Table 6.2.

n	$\hat{\gamma}$ (IS simulation)	RE	$\tilde{\gamma}$ (anal. approx.)	γ (numerical)
4	$6.808 \cdot 10^{-2}$	± 0.56 %	$6.5315 \cdot 10^{-2}$	$6.8055 \cdot 10^{-2}$
8	$5.891 \cdot 10^{-3}$	± 0.76 %	$5.8950 \cdot 10^{-3}$	$5.8878 \cdot 10^{-3}$
16	$4.553 \cdot 10^{-5}$	± 1.05 %	$4.6098 \cdot 10^{-5}$	$4.6047 \cdot 10^{-5}$
32	$2.852 \cdot 10^{-9}$	± 1.45 %	$2.8279 \cdot 10^{-9}$	$2.8271 \cdot 10^{-9}$
64	$1.055 \cdot 10^{-17}$	± 2.06 %	$1.0657 \cdot 10^{-17}$	$1.0656 \cdot 10^{-17}$
128	$1.490 \cdot 10^{-34}$	± 2.89 %	$1.5141 \cdot 10^{-34}$	$1.5141 \cdot 10^{-34}$

Table 6.2: Results for the $M/H_2/1$ queueing example.

Clearly, the probability estimate from simulation agrees with the numerical result. Also, the growth of the relative error from simulation is seen to agree with our theoretical bound, $\sqrt{n}$. For large n, the analytical approximation (6.10) turns out to be very accurate.

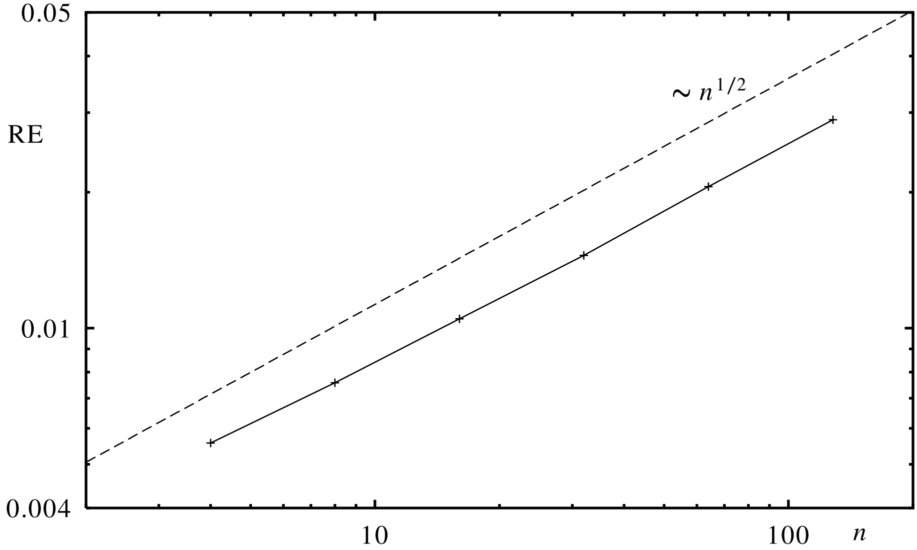


Figure 6.2: Relative error for the $M/H_2/1$ queueing example.

6.3.3 Signal detection example

Consider a constant false alarm rate (CFAR) radar detector which processes i.i.d. clutter returns from the background. The cell under test, denoted by Y, is compared to an adaptive threshold which is made up of a sum of n CFAR window samples $\{X_i\}_1^n$. For convenience, we assume that the threshold multiplier is set to unity. If Y exceeds the threshold a detection is declared. Given that there is no target in the test cell, the probability $\gamma = \mathbb{P}(Y > \sum_{i=1}^n X_i)$ represents the false alarm probability of the detector. We assume both Y and X_i to have a Weibull distribution with parameters $\beta = 2$ and $\lambda = 1$, as suggested in [Sri98b]. The results are shown in Table 6.3 and Figure 6.3.

The relative error asymptotically grows with $n^{1/4}$, again supporting our theoretical result. Also, there is a good agreement between the probability estimates from simulation and from the analytical approximation (6.10).

6.4 Conclusions

In this chapter, two methods for estimating rare event probabilities of the form $\mathbb{P}(\sum_{i=1}^n X_i < Y)$ for large n have been discussed.

The first method is based on importance sampling, using appropriate (exponential) tilting of the distributions of the random variables involved. Theoretically, we have shown that asymptotically the relative error for this simulation

n	$\hat{\gamma}$ (IS simulation)	RE	$\tilde{\gamma}$ (anal. approx.)
4	$3.033 \cdot 10^{-3}$	± 0.55 %	$2.999 \cdot 10^{-3}$
8	$1.376 \cdot 10^{-7}$	± 0.73 %	$1.351 \cdot 10^{-7}$
16	$1.313 \cdot 10^{-18}$	± 0.93 %	$1.303 \cdot 10^{-18}$
32	$2.099 \cdot 10^{-45}$	± 1.15 %	$2.112 \cdot 10^{-45}$
64	$1.368 \cdot 10^{-108}$	± 1.40 %	$1.398 \cdot 10^{-108}$
128	$3.437 \cdot 10^{-254}$	± 1.69 %	$3.469 \cdot 10^{-254}$
256	$6.470 \cdot 10^{-584}$	± 2.01 %	–

Table 6.3: Results for the signal detection example.

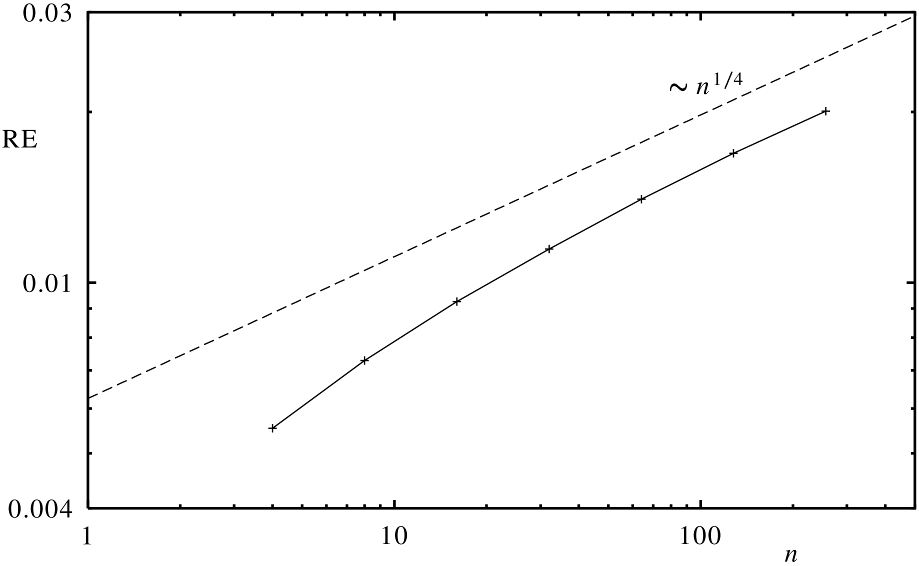


Figure 6.3: Relative error for the signal detection example.

decreases proportional to the square root of n (when Y has an exponential tail) or the fourth power root of n (when Y has a super-exponential tail), when keeping the number of replications fixed. This asymptotic behaviour has been confirmed experimentally. Furthermore, empirical results have shown that for smaller n, the relative error tends to grow a bit slower when Y has an exponential tail, and a bit faster in the other cases. Compared to the method for the estimation of such probabilities described in Chapter 5, the present method is more general since it is not limited to bounded Y distributions, and provides a slower growth of the relative error with n (except for the case of deterministic Y, where both methods are identical). However, in contrast to the method from Chapter 5, it can only be used in situations where the distribution of Y is known and can be tilted.

The second method in this chapter is based on analytical approximation, using the same (exponential) tilting of the random variables involved as in the first method. A proof has been given for its asymptotic validity (as $n \to \infty$) in the case when Y has an exponential or a super-exponential tail. Experimentally, the approximation was found to be quite good, with an error of typically a few percent at $n = 16$.

Applications in queueing systems, reliability models and signal detection have been considered.

6.A The analytical approximation

In this appendix, the validity of the analytical approximation presented in Section 6.2 is proved for a broad range of distributions of Y.

Lemma 6.1 *Given is a positive random variable Y with distribution function $F_Y(y)$ and moment generating function $\mathbb{E}e^{\theta Y}$ valid for any θ. Define the random variable Z with the following density:*

$$f_Z(z) = \rho_Z(1 - F_Y(z)),$$

where ρ_Z is the normalization constant such that $\int f_Z(z) = 1$.

Next, define the new random variables V and W by exponentially tilting the distributions of Y and Z, respectively, with a tilting parameter θ:

$$dF_V(z) = \rho_V e^{\theta z} dF_Y(z)$$

and

$$dF_W(z) = \rho_W e^{\theta z} dF_Z(z),$$

where ρ_V and ρ_W are normalization factors again.
Assuming that $\mathbb{E}V$ grows at most polynomially with θ, the following hold:

$$\mathbb{E}V - \mathbb{E}W = \mathcal{O}\left(\frac{1}{\theta}\right)$$

and

$$\mathrm{Var}\, W \le \frac{1}{\theta^2} + C\,\mathrm{Var}\, V$$

for some $C > 1$.

Proof: Denote by $\mu_Z(\cdot)$ the log moment generating function of Z:

$$\mu_Z(s) = \ln \rho_Z + \ln \int_0^\infty e^{sy}(1 - F_Y(y))dy.$$

Similarly, $\mu_Y(\cdot)$ denotes the log moment generating function of Y. Apply integration by parts to find for the latter:

$$\begin{aligned}
\mu_Y(s) &= \ln \int_0^\infty e^{sy} dF_Y(y) \\
&= \ln \left(-\int_0^\infty e^{sy} d(1 - F_Y(y)) \right) \\
&= \ln \left(-\Big[e^{sy}(1 - F_Y(y)) \Big]_0^\infty + \int_0^\infty s e^{sy}(1 - F_Y(y)) dy \right) \\
&= \ln \left(1 + s \int_0^\infty e^{sy}(1 - F_Y(y)) dy \right)
\end{aligned}$$

where the fact[5] that $\lim_{y\to\infty} e^{sy}(1 - F_Y(y)) = 0$ is used. To shorten the notation, define

$$J(s) = 1 + s \int_0^\infty e^{sy}(1 - F_Y(y)) dy,$$

so

$$\mu_Y(s) = \ln J(s)$$

and

$$\mu_Z(s) = \ln \rho_Z + \ln \frac{J(s) - 1}{s} = \ln \rho_Z - \ln s + \ln(J(s) - 1).$$

Note that the expectations of V and W at a tilting parameter θ are given by the derivatives of these log moment generating functions, $\mu_Y'(\theta)$ and $\mu_Z'(\theta)$, respectively. Thus:

$$\mathbb{E}V - \mathbb{E}W = \mu_Y'(\theta) - \mu_Z'(\theta) = \frac{J'(\theta)}{J(\theta)} + \frac{1}{\theta} - \frac{J'(\theta)}{J(\theta) - 1} = \frac{1}{\theta} - \frac{J'(\theta)}{J(\theta)(J(\theta) - 1)} = \frac{1}{\theta} - \frac{\mathbb{E}V}{J(\theta) - 1}.$$

Consider the last term: it is given that its numerator increases at most polynomially with θ, and its denominator increases at least exponentially fast[6] with θ. Therefore, this term vanishes compared to the $1/\theta$ term. This completes the proof of the first claim of the lemma.

Finally, consider the variance of W, omitting the argument (θ) of J for brevity:

$$\operatorname{Var} W = \mu_Z''(\theta) = \frac{1}{\theta^2} + \frac{J''(J - 1) - J'^2}{(J - 1)^2} \le \frac{1}{\theta^2} + \frac{J''J - J'^2}{(J - 1)^2},$$

where the last step uses the fact that $J'' \ge 0$, which is guaranteed because otherwise $\operatorname{Var} V$ would be negative. Using $\lim_{\theta\to\infty} J = \infty$, we find

$$\operatorname{Var} W \le \frac{1}{\theta^2} + C \frac{J''J - J'^2}{J^2} = \frac{1}{\theta^2} + C\mu_Y''(\theta) = \frac{1}{\theta^2} + C \operatorname{Var} V$$

[5]This follows from the fact that the moment generating function is valid for any θ: $e^{\theta y}(1 - F_Y(y)) = e^{\theta y} \int_y^\infty dF_Y(z) \le e^{-\theta y} \int_0^\infty e^{2\theta z} dF_Y(z)$, which for a given θ goes to zero as $y \to \infty$, since the integral is independent of y.

[6]Observe that for any $\epsilon > 0$, one has $J(s) \ge s \int_0^\epsilon e^{sy} a \, dy = a(e^{s\epsilon} - 1)$ with $a = 1 - F_Y(\epsilon)$. If ϵ is sufficiently small, $a > 0$, thus demonstrating exponential growth of $J(s)$.

for some $C > 1$, which completes the proof of the second claim of the lemma. $\square$

Theorem 6.1 *Given is a set of i.i.d. positive random variables X_i that satisfy the conditions of Theorem 5.2, and another independent positive random variable Y whose distribution has an exponential or super-exponential tail. Make the following definitions:*

$$S = \sum_{i=1}^{n} X_i,$$

$$\mu(s) = \ln \int_0^\infty e^{sx} dF_X(x).$$

The probability that $S < Y$ is given by

$$P_1 = \int_0^\infty \big(1 - F_Y(y)\big)\, dF_S(y),$$

which can be approximated by

$$\begin{aligned} P_2 &= e^{n\mu(-\theta)} \int_0^\infty \big(1 - F_Y(y)\big)\, e^{\theta y} \mathfrak{n}_{n\mu'(-\theta), n\mu''(-\theta)}(y) dy \\ &= \frac{e^{n\mu(-\theta)}}{\sqrt{2\pi n \mu''(-\theta)}} \int_0^\infty \big(1 - F_Y(y)\big)\, e^{\theta y} e^{-\frac{(y - n\mu'(-\theta))^2}{2n\mu''(-\theta)}} dy, \end{aligned}$$

with the tilting parameter θ chosen such that (as in Section 6.1.1)

$$n\mathbb{E}^* X_i = \mathbb{E}^* Y. \tag{6.11}$$

Finally, if

$$\frac{\operatorname{Var}^* Y}{\mathbb{E}^* Y} = \mathcal{O}(\theta^{-1}), \tag{6.12}$$

then P_2 converges to P_1 in the sense that

$$\lim_{n\to\infty} \frac{P_1}{P_2} = 1.$$

Proof:

Case 1: Y has an exponential tail

As in Section 6.1.2, we assume for simplicity that Y is exponentially distributed with parameter ν. Then the probability P_1 ($\mathbb{E}I$ in Section 6.1.2) can be calculated explicitly (note that by definition $\mu(\theta) = \ln(M_X(\theta))$):

$$P_1 = e^{n\mu(-\nu)}.$$

From (6.5), we have

$$\nu - \theta = \frac{1}{n\mu'(\theta)}.$$

Use this to rewrite the approximation P_2 as follows:

$$\begin{aligned}
P_2 &= e^{n\mu(-\theta)} \int_0^{\infty} e^{\theta y} e^{-\nu y} \mathfrak{n}_{n\mu'(-\theta),n\mu''(-\theta)}(y)dy \\
&= e^{n\mu(-\theta)} \int_0^{\infty} e^{-y/(n\mu'(-\theta))} \mathfrak{n}_{n\mu'(-\theta),n\mu''(-\theta)}(y)dy \\
&= e^{n\mu(-\theta)} \int_{-n\mu'(-\theta)}^{\infty} e^{-\frac{\sqrt{\mu''(-\theta)}}{\mu'(-\theta)\sqrt{n}}z-1} \mathfrak{n}_{0,1}(z)dz \\
&\approx e^{n\mu(-\nu)} e^{n\mu'(\theta)\cdot(-\nu+\theta)} \int_{-\infty}^{\infty} e^{-1} \mathfrak{n}_{0,1}(z)dz \\
&= e^{n\mu(-\nu)+1} e^{-1} = e^{n\mu(-\nu)} = P_1
\end{aligned}$$

as was to be shown. At the approximate-equals sign in the above, three approximations are made simultaneously. First, $n\mu(-\theta)$ is approximated using a Taylor series as $n\mu(-\nu) + n\mu'(-\theta)\cdot(-\nu+\theta)$. Second, the lower limit of the integral is replaced by its limit for $n \to \infty$; this extension of the range of the integral is allowed since the integrand in this region quickly goes to zero. Third, the factor $e^{-\frac{\sqrt{\mu''(-\theta)}}{\mu'(-\theta)\sqrt{n}}z}$ in the integrand is replaced by its limit for $n \to \infty$.

Case 2: Y has a super-exponential tail

First, define X_i^* as the exponentially tilted versions of X_i:

$$dF_X^*(x) = e^{-\mu(-\theta)} e^{-\theta x} dF_X(x),$$

where $e^{-\mu(-\theta)}$ is the normalization factor. Let $S^* = \sum_{i=1}^{n} X_i^*$ and rewrite P_1 as follows:

$$P_1 = e^{n\mu(-\theta)} \int_0^{\infty} (1 - F_Y(y)) e^{\theta y} dF_S^*(y).$$

Now note that P_2 has the same form as P_1 in the above, except for replacing the actual distribution of S^* by a normal density with the same mean and variance. The difference between P_1 (the true value of the probability) and P_2 (the approximation using a normal density) can be upper bounded as follows:

$$\begin{aligned}
|P_1 - P_2| &= e^{n\mu(-\theta)} \int_0^{\infty} (1 - F_Y(y))\, e^{\theta y} \, |f_S^* - \mathfrak{n}_{n\mu'(-\theta),n\mu''(-\theta)}(y)| \, dy \\
&\leq a(n) \cdot \frac{\theta}{\sqrt{n(r+1)}} \cdot e^{n\mu(-\theta)} \rho(\theta), \qquad (6.13)
\end{aligned}$$

where

$$\rho(\theta) = \int_0^{\infty} (1 - F_Y(y)) e^{\theta y} dy$$

and where $a(n)$ goes to zero when $n \to \infty$, according to Theorem 5.2 (the applicability of this theorem can be shown in the same way as in Section 6.1.2).

Next, we derive a lower bound on P_2. Define a random variable W with the density

$$f_W(y) = \frac{1}{\rho(\theta)}(1 - F_Y(y))e^{\theta y}.$$

This can be interpreted as an exponentially tilted version of a density proportional to $1 - F_Y(y)$. Using this (and omitting the subscripts of $\mathfrak{n}$ for brevity) P_2 can be expressed as

$$P_2 = e^{n\mu(-\theta)}\rho(\theta)\int_0^\infty \frac{1}{\rho(\theta)}(1 - F_Y(y))e^{\theta y}\,\mathfrak{n}_{\ldots}(y)dy = e^{n\mu(-\theta)}\rho(\theta)\int_0^\infty f_W(y)\,\mathfrak{n}_{\ldots}(y)dy.$$

Denote the standard deviation of W by σ and its mean by m. Then according to Chebyshev's inequality

$$\int_{m-c\sqrt{n\mu''(-\theta)}}^{m+c\sqrt{n\mu''(-\theta)}} f_W(y)dy \geq 1 - \frac{\sigma^2}{c^2 n\mu''(-\theta)},$$

for any positive constant c. Using this, we can lower bound P_2:

$$P_2 \geq e^{n\mu(-\theta)}\rho(\theta)\left(1 - \frac{\sigma^2}{c^2 n\mu''(-\theta)}\right)\min_{|y-m|\leq c\sqrt{n\mu''(-\theta)}}\mathfrak{n}_{\ldots}(y). \tag{6.14}$$

To estimate the minimum of the Gaussian density ($\mathfrak{n}$) in the above, note that by the tilting in (6.11), its peak is at $\mathbb{E}^*Y$; and by Lemma[7] 6.1, $\mathbb{E}W - \mathbb{E}^*Y = \mathcal{O}(1/\theta)$. So the Gaussian density in the above is evaluated at $c\sqrt{n\mu''(-\theta)} + \mathcal{O}(\theta^{-1})$ from its mean. Since its variance is $n\mu''(-\theta) = n(r+1)/\theta^2 + o(\theta^{-2})$, we have

$$\min_{|y-m|\leq c\sqrt{n\mu''(-\theta)}]}\mathfrak{n}_{\ldots}(y) = \frac{1}{\sqrt{2\pi n\mu''(-\theta)}}e^{-(c+\mathcal{O}(1)/\sqrt{n})^2} \approx \frac{e^{-c^2}}{\sqrt{2\pi n(r+1)/\theta^2}},$$

for large n. Substituting this into (6.14) yields

$$P_2 \geq e^{n\mu(-\theta)}\rho(\theta)\left(1 - \frac{\sigma^2}{c^2 n\mu''(-\theta)}\right)\frac{e^{-c^2}}{\sqrt{2\pi n(r+1)/\theta^2}}.$$

Finally, we compare the upper bound on $|P_1 - P_2|$ from (6.13) with the above lower bound on P_2:

$$\frac{|P_1 - P_2|}{P_2} \leq \frac{a(n)\sqrt{2\pi}}{e^{-c^2}\left(1 - \frac{\sigma^2}{c^2 n\mu''(-\theta)}\right)} \approx \frac{a(n)\sqrt{2\pi}}{e^{-c^2}\left(1 - \frac{\sigma^2}{c^2 n(r+1)/\theta^2}\right)}. \tag{6.15}$$

Using the second claim of Lemma 6.1, we calculate

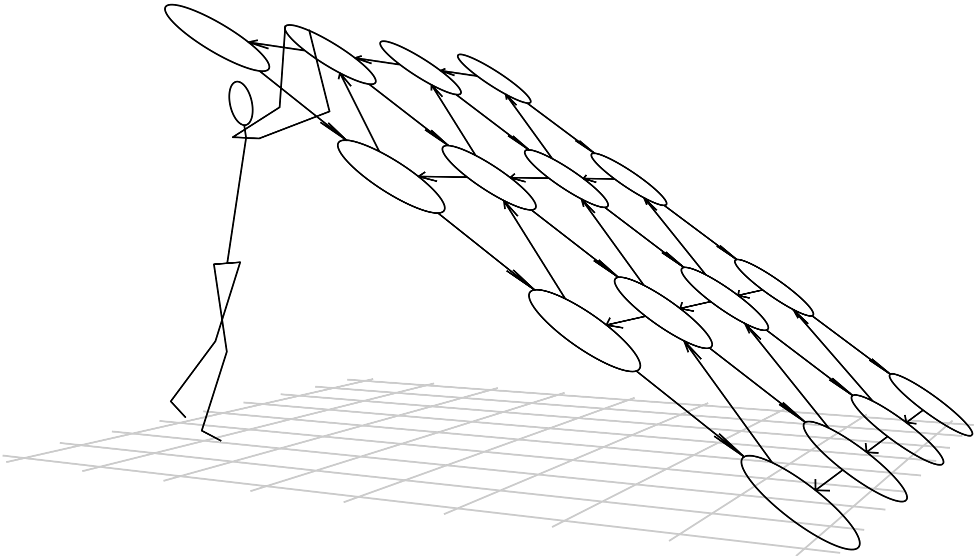
$$\frac{\sigma^2}{n/\theta^2} = \frac{\theta^2 \operatorname{Var} W}{n} \leq \frac{1}{n} + C\frac{\theta^2 \operatorname{Var}^* Y}{n} = \mathcal{O}(1),$$

[7] Using this lemma requires that $\mathbb{E}^*Y$ grow at most polynomially with θ, which follows from (6.12) in the same way as it did in Section 6.1.2

where the last step uses (6.8) from Section 6.1.2. Therefore, the denominator of the right-hand side of (6.15) is lower bounded by a constant, which can be made positive by choosing c large enough. Since the numerator $a(n)$ vanishes as $n \to \infty$, the relative difference between P_1 and P_2 vanishes as $n \to \infty$, implying that

$$\lim_{n\to\infty} \frac{P_1}{P_2} = 1.$$

QED. □



Chapter 7

Adaptive importance sampling simulation with state-independent tilting

In the previous chapters, importance sampling simulation methods have been described for several problems, and for these problems, the methods have been shown to be asymptotically efficient. In the literature, such asymptotic efficiency proofs are available for many problems. Unfortunately, when a simulation practitioner is confronted with a problem that cannot be transformed into one of those known problems, these results are useless to him. Devising a good importance sampling procedure for a new problem can be difficult, and analytically verifying its correctness (in the sense that it provides finite variance and thus a reliable estimate) or asymptotic efficiency is often complicated and time-consuming.

Because of these problems, a new approach to importance sampling has been tried in recent years: *adaptive* importance sampling. Adaptive here means[1] that in the course of the simulation procedure, the simulation parameters (change of measure) are adapted to the rare event of interest. Typically, adaptive importance sampling procedures perform several iterations, each consisting of simulating a (large) number of replications; after each iteration, the simulation parameters are changed based on the results up to then, to approach a setting which gives minimal variance of the resulting estimate of the rare-event probability of interest.

Several adaptive algorithms have been proposed in the literature. In [DT93b], the minimum variance along a line in the parameter space is searched for by

[1]Note that in [Hee98b], the term "adaptive importance sampling" is used for something that in this thesis would be called a "state-dependent change of measure"; see Chapter 8.

running simulations at several points along that line. In [DT93a], mean field annealing is used to find the minimum in a multi-dimensional parameter space. Simulations at one point in the parameter space can also be used to estimate the derivative (gradient) of the variance w.r.t. the parameters at that point. This is used in stochastic gradient descent techniques to step towards smaller variance [AQDT95]. In the stochastic Newton's method [RB00] not only the first but also the second derivatives are used to approach the minimum variance more effectively. Finally, the methods described in [Rub97], [RM98], [Lie99] and [LR00] search for the minimum variance by exploiting the fact that in importance sampling simulation, the samples taken at one point in the parameter space can be used to directly estimate the variance at other points in the parameter space. The latter method requires neither multiple simulations per iteration, nor the estimation of derivatives (which may be difficult to do accurately).

In this chapter, we will focus on the algorithm proposed in [RM98]. In Section 7.1, this algorithm will be described in more detail. Two improvements are subsequently described: a formulation based on cross-entropy in Section 7.2, and some improvements specific for Markovian systems in Section 7.3. Section 7.4 contains experimental results, applying the methods discussed to several simple queueing networks. Note that Section 7.1 and part of Section 7.2 consist of material that was already known from [RM98] and [Lie99]; it is included here for completeness, since the rest of this and the following chapter are based on it.

We will assume throughout that we are interested in (overflow) probabilities of the following form: the probability of reaching some (rare) "overflow" state before reaching an "absorbing" state, starting from a given initial state. Note that in principle, this is not a regenerative simulation. A typical application is a queueing network in which the overflow occurs when the total network population reaches some high level, the absorbing state is the state in which the network is empty, and the starting state is the state just after the first arrival to the empty system. In such cases, it is actually a regenerative simulation, and the results could also be used for the estimation of the steady state probability of the overflow event.

7.1 Principles

7.1.1 Preliminaries

Before the iterative procedure [RM98] for adapting the simulation parameters can be described, a few definitions need to be made. Note that this notation is not identical to the notation in [RM98], partially for consistency with the rest of the thesis, partially in an attempt to clarify the notation.

Define $\boldsymbol{v}$ to represent the change of measure (tilting). In general, there can be several tilting parameters, so $\boldsymbol{v}$ is a vector. Standard simulation (no tilting) is defined to correspond to $\boldsymbol{v} = 0$. Furthermore, $\boldsymbol{v}^*$ is the value of $\boldsymbol{v}$ that minimizes the variance of the estimator of the rare-event probability of interest.

Denote the sample path of one replication as Z, and the sample path for the ith replication as Z_i. The following functions of Z can be defined:

First, $I(Z)$ is the indicator function of the rare event. So $I(Z)$ is 1 if and only if the rare event occurs during Z.

Second, $f(Z, \boldsymbol{v})$ is the probability (or, for continuous systems, the probability density) of the sample path Z under the tilting $\boldsymbol{v}$. Then $f(Z, 0)$ is the probability (density) of Z in the original (untilted) system.

Third, $L(Z, \boldsymbol{v})$ is the likelihood ratio associated with the sample path Z and the tilting vector $\boldsymbol{v}$:

$$L(Z, \boldsymbol{v}) = \frac{f(Z, 0)}{f(Z, \boldsymbol{v})}.$$

Finally, define $\mathbb{E}_{\boldsymbol{v}}$ to denote the expectation under the tilting $\boldsymbol{v}$.

Based on simulation of N replications, with tilting $\boldsymbol{v}$, the importance sampling estimate of the rare event probability is given by

$$\hat{p} = \frac{\sum_{i=1}^{N} L(Z_i, \boldsymbol{v}) I(Z_i)}{N}$$

and its variance by

$$\hat{\sigma}^2 = \frac{1}{N-1} \left(\frac{\sum_{i=1}^{N} L^2(Z_i, \boldsymbol{v}) I(Z_i)}{N} - \hat{p}^2 \right). \tag{7.1}$$

Thus, in order to minimize this variance, one should find

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}} \mathbb{E}_{\boldsymbol{v}} I(Z) L^2(Z, \boldsymbol{v}).$$

Note that this equals

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}} \mathbb{E}_0 I(Z) L(Z, \boldsymbol{v}) \tag{7.2}$$

which is the minimization of the expected value of $I(Z)L(Z, \boldsymbol{v})$ in the original (untilted) system.

7.1.2 The variance-minimization procedure

A practical adaptive simulation experiment would start with choosing an initial value for $\boldsymbol{v}$, denoted by $\boldsymbol{v}_1$; an obvious choice would be $\boldsymbol{v}_1 = 0$, i.e., start with standard simulation. Subsequently a simulation involving say N replications would be performed. Based on the results of this, a new value $\boldsymbol{v}_2$ for the tilting vector would be calculated. This process is repeated, with the intention that the $\boldsymbol{v}_j$ converge to $\boldsymbol{v}^*$.

The algorithms used here for adapting $\boldsymbol{v}$ to minimize the variance are based on the interesting property of importance sampling simulation, that one set of samples can be used to estimate a quantity of interest for several values of the system parameters. In particular, with a set of samples corresponding to one value of $\boldsymbol{v}$, one can estimate the variance (7.1) that would be achieved for several other values of $\boldsymbol{v}$. In fact, one can perform a minimization: given a set of samples corresponding to one value of $\boldsymbol{v}$, one can calculate the value of $\boldsymbol{v}$ that minimizes the variance. This new value of $\boldsymbol{v}$ can then be used for the next simulation experiment.

We have already seen in (7.2) that minimal variance will be achieved in an importance sampling simulation with the tilting $\boldsymbol{v}^*$ that minimizes $\mathbb{E}_0 I(Z)L(Z,\boldsymbol{v})$. This can be rewritten as:

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}} \, \mathbb{E}_0\, I(Z)\, L(Z,\boldsymbol{v}) = \arg\min_{\boldsymbol{v}} \, \mathbb{E}_{\boldsymbol{v}_j}\, I(Z)\, L(Z,\boldsymbol{v}_j)\, L(Z,\boldsymbol{v}).$$

The latter form can be approximated by a sum over N samples Z_i taken from a distribution tilted by $\boldsymbol{v}_j$, thus yielding an approximation to $\boldsymbol{v}^*$. This approximation is used as the tilting for the next (i.e., $j+1$th) iteration:

$$\boldsymbol{v}_{j+1} = \arg\min_{\boldsymbol{v}} \sum_{i=1}^{N} I(Z_i)\, L(Z_i,\boldsymbol{v}_j)\, L(Z_i,\boldsymbol{v}). \tag{7.3}$$

Note that this expression contains two likelihood ratio factors, for fundamentally different reasons: $L(Z_i,\boldsymbol{v}_j)$ “compensates” for the fact that the samples Z_i are drawn from a $\boldsymbol{v}_j$-tilted distribution, whereas $L(Z_i,\boldsymbol{v})$ is part of the quantity $I(Z_i)\, L(Z_i,\boldsymbol{v})$ whose expectation (under the untilted distribution) we are trying to minimize.

Actually finding the minimum in (7.3) is not trivial: usually the $L(Z,\boldsymbol{v})$ function is non-linear, making this a non-linear optimization problem, which can only be solved by a time-consuming numerical procedure. In some cases, the minimization takes more time than the simulation itself. Also, a lot of memory may be needed to store state information from all N replications. In Section 7.2, a different approach for choosing $\boldsymbol{v}_{j+1}$ will be discussed, which overcomes these problems.

7.1.3 Changing the target event

The above minimization procedure assumes that the rare event is reached in at least some of the N replications that are simulated. Otherwise, $I(Z_i)$ would be 0 for all i, making the minimization (7.3) meaningless.

However, typically one would start with $\boldsymbol{v}_1 = 0$, i.e., initially standard simulation is used. Since the event of interest is rare, this would mean that it will not be observed, making the variance minimization impossible. In such cases, the rare event should be temporarily replaced by a less rare event (e.g. by lowering the overflow level in the case of a queueing system simulation). In practice, a good choice is such that the new target event is reached in about 1 percent of all replications (of the current iteration).

For the next iteration, the tilting parameter will typically favor the rare event, so another target event closer or identical to the rare event of interest can be chosen.

The above can be described mathematically as follows. Assume that some target function $U(Z)$ exists, and that the occurrence of the rare event of interest corresponds to $U(Z) \geq u_0$. For example, $U(Z)$ could be the highest level some buffer reaches on the sample path Z, and u_0 the overflow level of that buffer. A less rare event would then be $U(Z) \geq u$ with $u < u_0$.

7.1.4 Algorithm

The above procedures are summarized in the following algorithm.

1. Initialize as follows:
   $j := 1$ (iteration counter)
   $\boldsymbol{v}_1 := 0$ (initial tilting = no tilting, i.e., standard simulation)
2. Simulate N replications with tilting $\boldsymbol{v}_j$, yielding $Z_1 \ldots Z_N$.
3. Choose the overflow level u such that for a certain fraction (e.g., 1%) of all samples Z_i, the event $U(Z_i) \geq u$ happens. If $u > u_0$, set $u := u_0$.
4. Define the rare-event indicator $I(Z_i) = 1_{U(Z_i) \geq u}$, and find the new tilting factor $\boldsymbol{v}_{j+1}$ from the minimization (7.3).
5. Increment j and repeat steps 2–4, until $u = u_0$ and the tilting vector has converged (i.e., $\boldsymbol{v}_{j+1} \approx \boldsymbol{v}_j$).

Note that when applied to a queueing system, typically the tilted system is unstable after the first iteration, causing every overflow level to be reached with a high probability. In that case, step 3 will of course always set $u = u_0$ after the first iteration.

7.2 Cross-entropy formulation

In the previous section, a method for approaching the optimal tilting vector $\boldsymbol{v}^*$ has been described which, at every iteration, performs a numerical minimization of the estimated variance. In this section, a different approach to finding the optimal tilting is described, which requires much less computational effort.

It is well-known that, in principle, importance sampling simulation will provide an estimator with zero variance if the "ideal" change of measure is applied. This ideal change of measure is such that the simulation distributions of the random variables involved are precisely the original distributions conditioned on the occurrence of the rare event.

So instead of choosing $\boldsymbol{v}$ by explicitly minimizing the variance, one could also choose $\boldsymbol{v}$ by minimizing the "distance" between the $\boldsymbol{v}$-tilted distribution and the distribution conditioned on the occurrence of the rare event. In the iterative scheme, observations from the jth iteration could be used to estimate the conditional distribution, and then $\boldsymbol{v}_{j+1}$ could be chosen such that the $\boldsymbol{v}_{j+1}$-tilted distribution is as "close" as possible to the estimated conditional distribution. Of course, an exact definition of distance needs to be given. One possibility is the Kullback-Leibler cross-entropy, as proposed in [Lie99], [Rub99] and [LR00].

7.2.1 Theory

The Kullback-Leibler cross-entropy between two probability distributions $f(z)$ and $g(z)$ is defined as follows:

$$CE = \int f(z) \ln \frac{f(z)}{g(z)} dz.$$

Clearly, if the distributions $f(z)$ and $g(z)$ are identical, $CE = 0$; otherwise, $CE > 0$ (proof: see p. 156 in [KK92]). Note that this distance measure is not symmetric: in general, exchanging f and g in the above will result in a different value of CE.

We want to apply the Kullback-Leibler cross-entropy to measure the distance between the distribution to be used for the simulation (assumed to be of the form $f(z, \boldsymbol{v})$), and the ideal distribution, and then minimize it. To do this, substitute $g(z) = f(z, \boldsymbol{v})$ (i.e., the distribution to be optimized by changing $\boldsymbol{v}$), and $f(z) = \rho_0 I(z) f(z, 0)$ with $\rho_0 = \int I(z) f(z, 0) dz$; then $f(z)$ is the original distribution conditioned on the rare event (i.e., the "ideal" distribution). Thus, we need to do the following minimization:

$$\begin{aligned}
\boldsymbol{v}^{\dagger} &= \arg\min_{\boldsymbol{v}} \int \rho_0 I(z) f(z, 0) \ln \frac{\rho_0 I(z) f(z, 0)}{f(z, \boldsymbol{v})} dz \\
&= \arg\max_{\boldsymbol{v}} \int I(z) f(z, 0) \ln f(z, \boldsymbol{v}) dz \\
&= \arg\max_{\boldsymbol{v}} \mathbb{E}_0 I(Z) \ln f(Z, \boldsymbol{v}),
\end{aligned} \tag{7.4}$$

where $\boldsymbol{v}^{\dagger}$ denotes the value of $\boldsymbol{v}$ that minimizes the cross-entropy. In the above form the equation is not useful, since we do not know $\mathbb{E}_0 I(Z) \ln f(Z, \boldsymbol{v})$. However, we can rewrite it as follows:

$$\boldsymbol{v}^{\dagger} = \arg\max_{\boldsymbol{v}} \mathbb{E}_{\boldsymbol{v}_j} I(Z) L(Z, \boldsymbol{v}_j) \ln f(Z, \boldsymbol{v}).$$

This form can easily be approximated by a sum over N samples from the jth simulation, thus yielding an approximation to $\boldsymbol{v}^\dagger$ which we call $\boldsymbol{v}_{j+1}$:

$$\boldsymbol{v}_{j+1} = \arg\max_{\boldsymbol{v}} \sum_{i=1}^{N} I(Z_i)L(Z_i,\boldsymbol{v}_j)\ln f(Z_i,\boldsymbol{v}), \tag{7.5}$$

where the Z_i are drawn from a distribution tilted by $\boldsymbol{v}_j$. The above can be used instead of (7.3) in step 4 of the algorithm from Section 7.1.4.

Note that in [Lie99], a slightly different cross-entropy formulation is used: the $L(Z_i,\boldsymbol{v}_j)$ factor is omitted in (7.5). Since experiments in a few trial cases showed that omitting the factor typically gives worse variance and sometimes convergence problems, it has not been considered further.

7.2.2 Computational advantage

A drawback of the variance minimization algorithm from Section 7.1.2, is that it requires a lot of memory (to store all the state information from the simulation) and a lot of computations to (numerically) search for the optimal $\boldsymbol{v}_{j+1}$. In contrast, the cross-entropy method needs a much simpler computation for finding $\boldsymbol{v}_{j+1}$, at least if exponential tilting of the random variables is used. This is demonstrated in the following.

Single exponentially tilted random variable

Consider a very simple system, in which every replication consists of drawing just one sample from a random variable with density $g(x)$. In that case, the sample path Z is entirely given by that single sample. Assuming that exponential tilting is applied, the family of tilted distributions can be defined as follows:

$$f(x,v) = \frac{g(x)e^{vx}}{\rho(v)},$$

where $\rho(v)$ is the normalization factor:

$$\rho(v) = \int_0^\infty g(x)e^{vx}dx.$$

Note that $\boldsymbol{v}$ has been replaced by v, since it is a scalar in this simple case. Substituting this form of $f(x,v)$ into (7.4) yields

$$\begin{aligned} v^\dagger &= \arg\max_v \mathbb{E}_0 I(Z)\big(\ln\ g(Z) + vZ - \ln \rho(v)\big) \\ &= \arg\max_v \mathbb{E}_0 I(Z)\left(vZ - \ln \int_0^\infty g(x)e^{vx}dx\right), \end{aligned}$$

where the ln $g(Z)$ term has been dropped because it is independent of v and thus does not influence the $\arg\max_v$. To find the maximum in the right hand side, set its derivative w.r.t. v to zero:

$$0 = \mathbb{E}_0 I(Z) \left(Z - \frac{\int_0^\infty x\ g(x) e^{vx} dx}{\int_0^\infty\ g(x) e^{vx} dx} \right) = \mathbb{E}_0 \big(I(Z) Z \big) - \mathbb{E}_0 I(Z) \cdot \mathbb{E}_v Z.$$

Next, divide by $\mathbb{E}_0 I(Z)$ to find:

$$\mathbb{E}_0 \big(Z \mid I(Z) = 1 \big) = \mathbb{E}_v Z. \tag{7.6}$$

In words: *the optimal tilting is such, that the (unconditional) expectation of Z under the tilted distribution is equal to its expectation under the original distribution but conditioned on the occurrence of the rare event.*

The left-hand side of (7.6) is not known, of course. But in the iterative scheme, we can estimate it from the previous iteration as follows:

$$\mathbb{E}_0 \big(Z \mid I(Z) = 1 \big) = \frac{\mathbb{E}_0 Z I(Z)}{\mathbb{E}_0\ I(Z)} = \frac{\mathbb{E}_{v_j} Z I(Z) L(Z, v_j)}{\mathbb{E}_{v_j}\ I(Z) L(Z, v_j)} \approx \frac{\sum_{i=1}^N Z_i\, I(Z_i) L(Z_i, v_j)}{\sum_{i=1}^N\ I(Z_i) L(Z_i, v_j)},$$

assuming N replications were used in the jth simulation. Thus v_{j+1} would be chosen such that

$$\mathbb{E}_{v_{j+1}} Z = \frac{\sum_{i=1}^N Z_i\, I(Z_i) L(Z_i, v_j)}{\sum_{i=1}^N\ I(Z_i) L(Z_i, v_j)}, \tag{7.7}$$

which is practical if $\mathbb{E}_{v_{j+1}} Z$ is an easily invertible function of v_{j+1}. If Z is exponentially distributed, this is obviously no problem. In other cases, one would first need to find an expression for $\mathbb{E}_\vartheta Z$, i.e., the expected value of Z if its distribution is tilted exponentially with parameter ϑ; such an expression is given in (6.4) in terms of the Laplace-Stieltjes transform of the probability distribution function of Z. Analytical inversion w.r.t. ϑ of this expression may not always be feasible, but numerical evaluation (i.e., given $\mathbb{E}_\vartheta Z$, numerically approximate ϑ) typically is.

Multiple exponentially tilted random variables

In a practical system, there will in general be several random variables, from each of which multiple samples may be taken during one replication. As demonstrated below, the simple form (7.7) still holds in such cases, with only slight and obvious modifications.

Assume that there are n independent random variables, labeled $1 \dots n$. During a particular replication, N_l samples are taken from the lth random variable, with values $Z_{l,1} \dots Z_{l,N_l}$. The tilting parameter for the lth random variable is $\boldsymbol{v}(l)$, which is the lth component of $\boldsymbol{v}$ (note that we don't use subscripts here to prevent confusion with the iteration number). The untilted density of the lth random variable is $g_l(\cdot)$. Then

$$f(Z,\boldsymbol{v}) = \prod_{l=1}^{n}\prod_{j=1}^{N_l} \frac{g_l(Z_{l,j})e^{\boldsymbol{v}(l)Z_{l,j}}}{\rho_l(\boldsymbol{v}(l))}$$

with

$$\rho_l(v) = \int_0^\infty g_l(x)e^{vx}dx.$$

Substitution into (7.4) yields

$$\begin{aligned}
\boldsymbol{v}^\dagger &= \arg\max_{\boldsymbol{v}} \mathbb{E}_0 I(Z) \ln \prod_{l=1}^{n}\prod_{j=1}^{N_l} \frac{g_l(Z_{l,j})e^{\boldsymbol{v}(l)Z_{l,j}}}{\rho_l(\boldsymbol{v}(l))} \\
&= \arg\max_{\boldsymbol{v}} \mathbb{E}_0 I(Z) \sum_{l=1}^{n}\sum_{j=1}^{N_l} \big(\ln\ g_l(Z_{l,j}) + \boldsymbol{v}(l)Z_{l,j} - \ln\rho_l(\boldsymbol{v}(l))\big) \\
&= \arg\max_{\boldsymbol{v}} \mathbb{E}_0 I(Z) \sum_{l=1}^{n} \left(-N_l \ln\rho_l(\boldsymbol{v}(l)) + \sum_{j=1}^{N_l} \boldsymbol{v}(l)Z_{l,j}\right).
\end{aligned}$$

Differentiate this w.r.t. $\boldsymbol{v}(l)$ to find the maximum:

$$\begin{aligned}
0 &= \mathbb{E}_0 I(Z)\left(-N_l \frac{\int_0^\infty x\ g_l(x)e^{\boldsymbol{v}(l)x}dx}{\int_0^\infty\ g_l(x)e^{\boldsymbol{v}(l)x}dx} + \sum_{j=1}^{N_l} Z_{l,j}\right) \\
&= \mathbb{E}_0\left(I(Z)\sum_{j=1}^{N_l} Z_{l,j}\right) - \mathbb{E}_0\big(N_l I(Z)\big)\cdot \mathbb{E}_{\boldsymbol{v}(l)}Z_l.
\end{aligned}$$

Proceeding as in the case of a single random variable, we find the following equation for choosing $\boldsymbol{v}_{j+1}(l)$:

$$\mathbb{E}_{\boldsymbol{v}_{j+1}(l)}Z = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N_l} Z_{l,j}\, I(Z)L(Z,\boldsymbol{v}_j)}{\sum_{i=1}^{N}\sum_{j=1}^{N_l}\ \ I(Z)L(Z,\boldsymbol{v}_j)}, \tag{7.8}$$

where i counts the replications, and N_l, Z and $Z_{l,j}$ belong to the ith replication. Practically, the double sum is just a sum over all samples from the lth random variable during all replications that reach the rare event.

As an example of the use of (7.8), consider the simulation of a queueing system with an exponential interarrival time distribution with rate λ. Exponentially tilting such a distribution boils down to changing the parameter λ to some other value λ_{j+1} (for the $j+1$th iteration), so the expectation of the tilted interarrival time distribution is $1/\lambda_{j+1}$. When using (7.8) to find the tilting of the arrival distribution, its left hand side would thus be the expectation $1/\lambda_{j+1}$, while the right-hand side is the average of all samples of the interarrival time drawn on sample paths leading to the rare event, weighted according to the likelihood ratio.

7.2.3 Comparison between minimization of variance and of cross-entropy

The optimization procedure described in Section 7.1.2 changes the tilting parameter $\boldsymbol{v}$ such that the variance of the resulting importance sampling estimator is minimized. Clearly, this is a useful property of an adaptive importance sampling scheme.

In the procedure described in Section 7.2.1, an abstract quantity, the cross-entropy, is minimized. It is not immediately clear, however, that this minimization procedure is useful for importance sampling simulation, since there is no clear relationship between the simulation variance and this cross-entropy. Still, as will be seen in Section 7.4, this minimization procedure applied in adaptive importance sampling simulation gives quite good results, at least for changes of measure involving exponential tilting. A possible explanation for this is that minimizing the cross-entropy between the simulation distribution and the conditional distribution brings the former closer to the latter, i.e., closer to the distribution that would give a zero-variance estimator; one should expect the variance to decrease while doing so.

In this section, a more analytical comparison between the two methods is made. We start by analyzing the variance-minimization method when used with exponential tilting.

First, define $f(x, v)$ as earlier:

$$f(x, v) = \frac{g(x)e^{vx}}{\rho(v)},$$

where $\rho(v)$ is the normalization factor:

$$\rho(v) = \int_0^\infty g(x)e^{vx}dx.$$

Substituting this into (7.2) yields

$$v^* = \arg\min_v \mathbb{E}_0 \frac{g(Z)}{g(Z)e^{vZ}\rho^{-1}(v)} I(Z) = \arg\min_v \mathbb{E}_0\, e^{-vZ}\rho(v)I(Z).$$

To find the minimum, differentiate w.r.t. v and set to zero:

$$0 = \mathbb{E}_0\, e^{-v^*Z}\rho(v^*)I(Z)\left(-Z + \frac{\rho'(v^*)}{\rho(v^*)}\right) = \mathbb{E}_0 L(Z, v^*)I(Z)\left(-Z + \frac{\int_0^\infty x\ g(x)e^{v^*x}dx}{\int_0^\infty\ g(x)e^{v^*x}dx}\right).$$

Division by $\mathbb{E}_0 I(Z)$ and rewriting yields

$$\frac{\mathbb{E}_0\left(Z\,L(Z,v^*)\mid I(Z)=1\right)}{\mathbb{E}_0\left(\quad L(Z,v^*)\mid I(Z)=1\right)}=\mathbb{E}_{v^*}Z. \tag{7.9}$$

For comparison, the corresponding cross-entropy result (7.6) is

$$\mathbb{E}_0\left(Z\mid I(Z)=1\right)=\mathbb{E}_{v^\dagger}Z.$$

Thus, both methods choose v such that the expectation of the v-tilted random variable becomes equal to the conditional expectation of Z; the only difference is that in the variance minimization method an additional weighting factor equal to the likelihood ratio[2] $L(Z,v)$ is used in this conditional expectation of Z.

Clearly, this difference will, in general, make $v^*\neq v^\dagger$. However, in the ideal case of zero variance, it is known (and easy to see) that, given the occurrence of the target event, $L(Z,v)$ becomes a constant. So if the zero variance change of measure is in the family of distributions $f(x,\boldsymbol{v})$, then both methods will find it (assuming some regularity).

An alternative for the variance minimization method

As was stated before, finding $\boldsymbol{v}^*$ which minimizes the simulation variance as given by (7.2) or (7.3), is not computationally simple. In the specific case of exponential tilting, (7.2) reduces to (7.9), which still cannot be solved easily due to the presence of $\boldsymbol{v}^*$ in a non-linear way on both sides of the equation.

However, with the following substitution one could simplify the use of (7.9): substitute v_j for v^* in the left-hand side, and v_{j+1} in the right-hand side. We get:

$$\frac{\mathbb{E}_0\left(Z\,L(Z,v_j)\mid I(Z)=1\right)}{\mathbb{E}_0\left(\quad L(Z,v_j)\mid I(Z)=1\right)}=\mathbb{E}_{v_{j+1}}Z.$$

This version has the same computational convenience as (7.8): the left-hand side can be computed directly on the basis of simulation results from the jth iteration, after which the right-hand side can be inverted to find v_{j+1}. This gives us yet another way (besides variance-minimization and cross-entropy) to choose the tilting vector for the next iteration in step 4 in the algorithm from Section 7.1.4.

What could be expected from using this method? When v_j is near the optimal value v^*, the above substitution of v_j for v^* introduces only a small error. In

[2]Note that this factor is the only factor that makes the left-hand side depend on v. This causes the computational complexity difference between the variance-minimization method and the cross-entropy method: with the cross-entropy method, the left-hand side can be calculated first, and then the right-hand side can be solved for v, whereas for the variance-minimization method an iterative procedure is needed.

that case, we might expect this method to converge indeed to the minimum variance. That would be an improvement over the cross-entropy method, which in principle does not converge to the minimum variance, and needs a comparable computational effort. On the other hand, if v_j is far from the optimal v^*, inserting this wrong value into the left-hand side of (7.9) could be expected to make the convergence harder.

Actually, in Section 7.4 it will be shown experimentally that the performance of this alternative method hardly differs from the variance-minimization and cross-entropy methods; see Table 7.1.

7.3 Adaptation to Markovian models

Many models used for queueing systems performance evaluation, are (or can be converted to) discrete-time Markov chains (DTMCs). This is the case if all inter-arrival times and service times have an exponential distribution, and the quantity of interest is, for example, an overflow probability. Note that some other performance measures, like delays, cannot be obtained from the discrete-time Markov chain description.

A DTMC for a Markovian queueing model has a highly regular structure. First of all, the states typically can be arranged conveniently on a grid with as many dimensions as the number of queues, with each coordinate representing the number of customers in one of the queues. Secondly, every transition in the DTMC corresponds to an elementary event in the queueing model: an arrival or a service completion at one of the queues. For convenience, we will henceforth refer to such events as "transition events". It should be noted that these transition events are defined independently of the state; i.e., there is only one transition event for a service completion at a given queue, and this single transition event corresponds to a transition out of every state in the DTMC in which this particular queue is non-empty. Third, we see that not all transition events are "enabled" in every state: e.g., in a state where a particular queue is empty, the service completion event of that particular queue is not possible, i.e., not enabled. Finally, with every transition event a rate is associated (namely the rate of the exponential inter-arrival or service time distribution in the continuous-time model), and the probability of a given transition out of a given state of the DTMC is that event's rate divided by the sum of the rates of all transition events that are enabled in that state.

As an example, consider a two-node tandem network as depicted in Figure 2.1. Evidently, the state of the network is completely described by n_1 and n_2: the numbers of customers in the queues. There are three transition events: the arrival at the first queue with rate λ, service completion at the first queue with

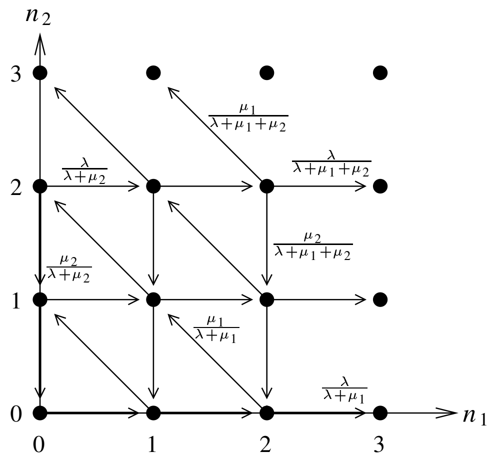


Figure 7.1: DTMC for two queues in tandem.

rate μ_1, and service completion at the second queue with rate μ_2. The state space is depicted as a two-dimensional grid in Figure 7.1, with arrows denoting the transitions. Note that in states on the horizontal axis (i.e., states where the second queue is empty) the transition event corresponding to service completion at the second queue is not enabled, so there is no arrow pointing down; similarly the service completion at the first queue is not enabled in the states on the vertical axis.

From the above, it follows that every transition probability in the DTMC is a (simple) function of the rates of the continuous-time model. For the simulation of overflows in the continuous-time queueing models, it is known that exponential tilting generally works well ([PW89], [Sad91], and our examples in Section 7.4), so it makes sense to try to apply the equivalence of exponential tilting to the DTMC. Exponential tilting of an exponential distribution boils down to changing the distribution's rate; so the equivalent tilting of the DTMC would be changing the rates, and letting the transition probabilities change according to their functional dependence on those rates.

7.3.1 Cross-entropy for DTMCs — theory

In order to apply the cross-entropy formulation, let us first build a mathematical description of one replication of a DTMC simulation, denoting its sample path by Z. Start by labelling all transition events (arrivals, service completions), starting from 1. Then define the following:

- λ_k is the rate of transition event k.

- n is the number of transitions in this sample path.
- c_j is the label of the transition event that occurred at the jth transition of the sample path.
- E_{jk} is an indicator: it is 1 if at the jth transition, transition event k was enabled, and 0 otherwise.
- I is the indicator function of the occurrence of the target (rare) event during this sample path.

Note that n, c_j, E_{jk} and I are random variables, since they are functions of the sample path. With the above definitions, the a-priori probability that at the jth transition, transition event l happens, is obviously given by $\lambda_l / \sum_k E_{jk}\lambda_k$. Therefore, the probability of the entire sample path Z is given by

$$\mathbb{P}(Z) = \prod_{j=1}^{n} \frac{\lambda_{c_j}}{\sum_k E_{jk}\lambda_k}.$$

The goal now is to find a set of transition rates λ_k which minimizes the cross-entropy as given by[3] (7.4). That is:

$$\boldsymbol{\lambda}^\dagger = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_0 I \ln \prod_{j=1}^{n} \frac{\lambda_{c_j}}{\sum_k E_{jk}\lambda_k} = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_0 I \sum_{j=1}^{n} \left(\ln \lambda_{c_j} - \ln \sum_k E_{jk}\lambda_k \right).$$

To find the maximum, differentiate this w.r.t. λ_l:

$$0 = \mathbb{E}_0 I \sum_{j=1}^{n} \left(\frac{1_{c_j=l}}{\lambda_l^\dagger} - \frac{E_{jl}}{\sum_k E_{jk}\lambda_k^\dagger} \right), \tag{7.10}$$

where $1_{c_j=l}$ is the indicator function which is 1 if and only if $c_j = l$, i.e., if at the jth transition, the transition event l happened. Equation (7.10) can be rewritten as

$$\mathbb{E}_0 I \sum_{j=1}^{n} 1_{c_j=l} = \mathbb{E}_0 I \sum_{j=1}^{n} \frac{E_{jl}\lambda_l^\dagger}{\sum_k E_{jk}\lambda_k^\dagger}, \tag{7.11}$$

or, equivalently,

$$\frac{1}{\mathbb{E}_0 I\, n} \mathbb{E}_0\, I \sum_{j=1}^{n} 1_{c_j=l} = \frac{1}{\mathbb{E}_0 I\, n} \mathbb{E}_0\, I \sum_{j=1}^{n} \frac{E_{jl}\lambda_l^\dagger}{\sum_k E_{jk}\lambda_k^\dagger}.$$

[3] In (7.4), the cross-entropy is minimized by changing the tilting vector $\boldsymbol{\nu}$. In the present case however, it is more convenient to refer to the rates themselves, and minimize the cross-entropy as a function of the tilted rates instead of the tilting vector. Formally, the tilting vector $\boldsymbol{\nu}$ could be defined as the difference between the tilted rates vector and the untilted rates vector, to show the equivalence.

The left-hand side obviously is the (observed) conditional probability of transition event l on sample paths leading to the rare event. The right hand side is a weighted average of the a-priori probability of transition event l on sample paths to the rare event, using the transition rates $\boldsymbol{\lambda}^\dagger$. Intuitively, setting these two equal seems like a good way for choosing the optimal tilting vector $\boldsymbol{\lambda}^\dagger$.

The set of equations (7.11) (note that there is one such equation for every transition event l) typically cannot be solved directly, since the expectations involved are hard to calculate. In Section 7.1 and 7.2, we have seen several cases where such an equation was used to express the tilting parameter vector for the next iteration in terms of simulation results from the previous iteration. Applied to (7.11), this would mean approximating the left-hand side using simulation results, and then solving the right-hand side w.r.t. $\lambda_k^\dagger$. Unfortunately, the right-hand side still contains an expectation over a sum over states of a sample path, which makes it rather hard to invert w.r.t. $\lambda_k^\dagger$. One could use the sample paths generated in the previous simulation to evaluate this expectation multiple times, in order to iteratively solve the right-hand side, but this is computationally intensive. Therefore, it is desirable to find a simpler way for approximately calculating the optimal tilting vector $\boldsymbol{\lambda}^\dagger$.

Let us rewrite (7.10) once more and multiply it by $\lambda_l^\dagger$, to find

$$0 = \mathbb{E}_0 I \sum_{j=1}^{n} \frac{1}{\sum_k E_{jk}\lambda_k^\dagger} \left(1_{c_j=l} \textstyle\sum_k E_{jk}\lambda_k^\dagger - \lambda_l^\dagger E_{jl} \right).$$

The right-hand side can be interpreted as a weighted average of the part between parentheses, where one of the weighting factors is $1/\sum_k E_{jk}\lambda_k^\dagger$. If we, as an approximation[4], leave out this weighting factor, we get:

$$0 = \mathbb{E}_0 I \sum_{j=1}^{n} \left(1_{c_j=l} \textstyle\sum_k E_{jk}\lambda_k^\ddagger - \lambda_l^\ddagger E_{jl} \right), \tag{7.12}$$

where the symbol $\ddagger$ is used instead of $\dagger$ to emphasize that the solution $\boldsymbol{\lambda}^\ddagger$ is not really the one that minimizes the cross-entropy. The great advantage of (7.12) over the more exact (7.11) is that it can be solved efficiently on the basis of simulation results, as explained in the next section.

Note that in the case where all transition events are enabled in all states, i.e., $E_{jk} = 1$ for all j and k, both versions (7.11) and (7.12) are equivalent. If this is not the case, then the difference is that observations from states in which one or more transitions are not enabled, weigh a bit heavier in (7.11) than they do in (7.12). An experimental comparison of both methods is given in Section 7.4.1;

[4] Actually, the author originally came up with something equivalent to (7.12) on heuristic grounds, experimentally found it to work well, and did not derive the close relationship with cross entropy minimization until later.

from the experimental results it is clear that the methods perform equally well, so this weighing difference apparently does not significantly affect the results.

7.3.2 Cross-entropy for DTMCs — practice

We now proceed to transform (7.12) into a simple linear matrix equation, which can be solved to find the (almost) optimal transition rates $\boldsymbol{\lambda}^{\ddagger}$ after replacing expectations by simulation results.

First, rewrite (7.12) as follows, basically taking the term corresponding to $k = l$ out of the summation over k:

$$0 = \mathbb{E}_0 I \sum_{j=1}^{n} \left(1_{c_j=l} \textstyle\sum_{k \neq l} E_{jk}\, \lambda_k^{\ddagger} - (1 - 1_{c_j=l}) E_{jl}\, \lambda_l^{\ddagger} \right). \tag{7.13}$$

Then define m_{lk} as follows:

$$m_{lk} \stackrel{\text{def.}}{=} \mathbb{E}_0 I \sum_{j=1}^{n} 1_{c_j=l} E_{jk}.$$

When divided by $\mathbb{E}_0 I$ (i.e., the rare event probability of interest), this quantity m_{lk} can be interpreted as the expected number of transitions at which transition event k is enabled and transition event l happens, during one replication which reaches the rare event. Next, define

$$\overline{m}_l \stackrel{\text{def.}}{=} \sum_{k \neq l} m_{kl} = \mathbb{E}_0 I \sum_{j=1}^{n} E_{jl} \sum_{k \neq l} 1_{c_j=k} = \mathbb{E}_0 I \sum_{j=1}^{n} E_{jl} (1 - 1_{c_j=l}),$$

where the last equals sign uses the fact that if c_j is not equal to any $k \neq l$, it must be equal to l.

Using m_{lk} and $\overline{m}_l$ as defined above, we can rewrite the entire set (for all l) of equations (7.13) as one matrix equation:

$$\begin{bmatrix} -\overline{m}_1 & m_{12} & m_{13} & \cdots \\ m_{21} & -\overline{m}_2 & m_{23} & \cdots \\ m_{31} & m_{32} & -\overline{m}_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{pmatrix} \lambda_1^{\ddagger} \\ \lambda_2^{\ddagger} \\ \lambda_3^{\ddagger} \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \tag{7.14}$$

Of course, the expectations in m_{kl} are generally not known, but we can approximate them by simulating many, say N, replications:

$$\hat{m}_{lk} = \frac{1}{N} \sum_{i=1}^{N} I_i L_i \sum_{j=1}^{n_i} 1_{c_{ij}=l} E_{ijk}, \tag{7.15}$$

where the extra index i in I_i, L_i, n_i c_{ij} and E_{ijk} indicates that they belong to the ith replication. In practice, the factor $1/N$ can be left out, since it is the same for all $\hat{m}_{lk}$ and only their ratios are relevant for the solution of (7.14).

Now we finally have all we need for applying the algorithm from Section 7.1.4 to DTMC models: we can use the above calculations in step 4 of the algorithm to calculate an approximation for the optimal tilting for the next iteration on the basis of simulation results from the previous iteration. Note that the matrix equation (7.14) can only determine the rates $\boldsymbol{\lambda}^{\ddagger}$ up to a constant factor, since its right-hand side is 0. However, this is enough, since in a DTMC simulation only the ratio between the rates is relevant. In practice (e.g., in tables in the experiments section), we normalize the rates such that their sum equals 1.

Finally, note that in spite of their somewhat non-straightforward definition (7.15), the quantities $\hat{m}_{lk}$ can be calculated quite simply by the simulation program, as follows. Start by setting all matrix elements to zero. For every replication (sample path) leading to the rare event, note the likelihood ratio L, then loop over all transitions of that sample path, and for every *enabled* transition event k add L to the matrix element m_{lk}, where l is the transition event that actually happened at that transition. After doing this, compute the diagonal elements such that the sum of every column is 0.

7.4 Experimental results

The methods described in the previous sections will now be put to the test. Several example queueing networks will be used for this. Whenever possible, non-simulation results will be used to verify the results from the adaptive importance sampling simulation methods.

7.4.1 Two Markovian queues in tandem with feedback

The first test case consists of two queues in tandem, where a fraction p of the output of the second queue is fed back to the input of the first queue. Furthermore, external arrivals occur at the first queue with an exponentially distributed interarrival time (rate λ), and the queues also have an exponentially distributed service time (with rates μ_1 and μ_2). This is illustrated in Figure 7.2. The rare event of interest is overflow of the total population of the two queues, i.e., $n_1 + n_2 \geq M$, for some large M.

We use the following parameters: $\lambda = 0.1$, $\mu_1 = 0.6$, $\mu_2 = 0.4$ and $p = 0.5$. Because of the feedback, the total arrival rate at the first queue is 0.2, causing the first queue to have a load of $0.2/0.6 = 0.333$ and the second queue of $0.2/0.4 = 0.5$. So both queues are stable. We set the overflow level M to 50.

Variance-minimization method (Section 7.1)

iteration	replications	λ	μ_1	μ_2	p	estimate	rel.std.dev.
1	10^3	0.1	0.6	0.4	0.5	0	∞
2	10^3	0.203	0.543	0.339	0.348	$3.710 \cdot 10^{-15}$	0.2600
3	10^3	0.198	0.593	0.300	0.327	$2.699 \cdot 10^{-15}$	0.0614
4	10^3	0.194	0.585	0.307	0.336	$2.857 \cdot 10^{-15}$	0.0445
5	10^3	0.199	0.592	0.304	0.331	$3.107 \cdot 10^{-15}$	0.1559
6	10^3	0.186	0.583	0.309	0.317	$2.531 \cdot 10^{-15}$	0.0714
7	10^3	0.198	0.597	0.308	0.337	$2.891 \cdot 10^{-15}$	0.0403
8	10^3	0.199	0.594	0.302	0.332	$2.383 \cdot 10^{-15}$	0.0409
9	10^3	0.198	0.590	0.302	0.330	$3.024 \cdot 10^{-15}$	0.1372
8	10^5	0.199	0.594	0.302	0.332	$2.670 \cdot 10^{-15}$	0.0049
9	10^5	0.198	0.589	0.306	0.333	$2.668 \cdot 10^{-15}$	0.0047
10	10^5	0.197	0.589	0.305	0.334	$2.672 \cdot 10^{-15}$	0.0046

Cross-entropy method (Section 7.2)

iteration	replications	λ	μ_1	μ_2	p	estimate	rel.std.dev.
1	10^3	0.1	0.6	0.4	0.5	0	∞
2	10^3	0.214	0.547	0.355	0.348	$2.013 \cdot 10^{-15}$	0.1636
3	10^3	0.206	0.579	0.310	0.340	$2.681 \cdot 10^{-15}$	0.0547
4	10^3	0.200	0.589	0.299	0.332	$2.408 \cdot 10^{-15}$	0.0436
5	10^3	0.200	0.598	0.302	0.331	$2.567 \cdot 10^{-15}$	0.0446
6	10^3	0.198	0.591	0.306	0.331	$2.778 \cdot 10^{-15}$	0.0400
7	10^3	0.200	0.595	0.305	0.328	$2.858 \cdot 10^{-15}$	0.0387
8	10^3	0.198	0.595	0.304	0.329	$2.688 \cdot 10^{-15}$	0.0433
8	10^5	0.198	0.595	0.304	0.329	$2.661 \cdot 10^{-15}$	0.0045

Alternative variance-minimization method (Section 7.2.3)

iteration	replications	λ	μ_1	μ_2	p	estimate	rel.std.dev.
1	10^3	0.1	0.6	0.4	0.5	0	∞
2	10^3	0.214	0.547	0.355	0.348	$2.013 \cdot 10^{-15}$	0.1636
3	10^3	0.199	0.591	0.301	0.343	$2.762 \cdot 10^{-15}$	0.0477
4	10^3	0.199	0.585	0.309	0.330	$2.771 \cdot 10^{-15}$	0.0471
5	10^3	0.196	0.607	0.309	0.332	$2.520 \cdot 10^{-15}$	0.0431
6	10^3	0.201	0.578	0.300	0.332	$2.530 \cdot 10^{-15}$	0.0452
7	10^3	0.204	0.617	0.308	0.336	$2.445 \cdot 10^{-15}$	0.0541
7	10^5	0.204	0.617	0.308	0.336	$2.635 \cdot 10^{-15}$	0.0058

DTMC method (Section 7.3)

iteration	replications	λ	μ_1	μ_2	p	estimate	rel.std.dev.
1	10^3	0.1	0.6	0.4	0.5	0	∞
2	10^3	0.187	0.497	0.316	0.335	$2.784 \cdot 10^{-15}$	0.1336
3	10^3	0.184	0.541	0.275	0.326	$2.651 \cdot 10^{-15}$	0.0367
4	10^3	0.181	0.542	0.277	0.334	$2.805 \cdot 10^{-15}$	0.0364
5	10^3	0.181	0.543	0.276	0.331	$2.761 \cdot 10^{-15}$	0.0384
6	10^3	0.181	0.542	0.276	0.334	$2.740 \cdot 10^{-15}$	0.0333
7	10^3	0.181	0.543	0.275	0.333	$2.719 \cdot 10^{-15}$	0.0391
7	10^5	0.181	0.543	0.275	0.333	$2.678 \cdot 10^{-15}$	0.0037

continued on next page

Table 7.1: Experimental results for the tandem queue with feedback.

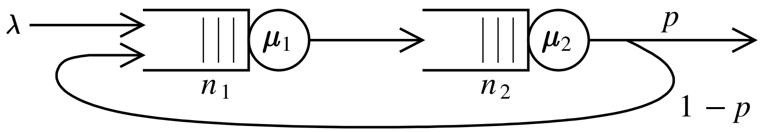


Figure 7.2: Two queues in tandem with feedback.

continued from previous page

"Precise" DTMC method (Section 7.3, eq. (7.11))

iteration	replications	λ	μ_1	μ_2	p	estimate	rel.std.dev.
1	10^3	0.1	0.6	0.4	0.5	0	∞
2	10^3	0.186	0.484	0.329	0.336	$2.019 \cdot 10^{-15}$	0.2271
3	10^3	0.174	0.540	0.286	0.320	$2.700 \cdot 10^{-15}$	0.0510
4	10^3	0.183	0.542	0.275	0.331	$2.681 \cdot 10^{-15}$	0.0376
5	10^3	0.184	0.539	0.276	0.332	$2.673 \cdot 10^{-15}$	0.0392
6	10^3	0.183	0.539	0.278	0.330	$2.553 \cdot 10^{-15}$	0.0367
7	10^3	0.183	0.543	0.274	0.331	$2.595 \cdot 10^{-15}$	0.0322
7	10^5	0.183	0.543	0.274	0.331	$2.660 \cdot 10^{-15}$	0.0038

Since this is a Markovian system, all methods discussed in this chapter can be applied to it. The results are presented in Table 7.1. For each of the methods, this table shows simulation parameters and results (estimate and error as fraction of the estimate) for a number of iterations, starting with the untilted system. For most iterations 10^3 replications were used; after convergence, 10^5 regenerations were used to check whether the relative error decreases appropriately (i.e., by a factor of $\sqrt{10^5/10^3} = 10$).

This problem is amenable to the analytical/numerical method discussed in Chapter 2, which gives the exact (except for numerical inaccuracies) probability as $2.6645 \cdot 10^{-15}$. Comparing this with the estimates and their relative standard deviations in the table shows that the simulation estimates are good: in most cases, the difference with the exact answer is less than one standard deviation.

In [FLA91], a method is discussed to analytically choose the simulation parameters for simulation of overflows in a Jackson network, based on large deviations heuristics. Applied to the present problem, this method gives: $\lambda^* = 0.2$; $\mu_1^* = 0.6$; $\mu_2^* = 0.3$; $p^* = 0.3333$. From the tables it is clear, that all four adaptive importance sampling algorithms also converge to these values. Note that for checking this for the DTMC methods, the above rates must be normalized first, because in DTMC simulations only the ratios between the rates are relevant.

In Section 7.2.3, it was noted that in principle, the cross-entropy method need not converge to a simulation with minimal variance, as opposed to the variance-minimization method. Thus one would expect that the cross-entropy method ends up with a larger variance. However, the table shows that both methods give approximately equal variance, and in fact it seems that the cross-entropy method is slightly better. Apparently then, the disadvantage of the cross-entropy method (i.e., the fact that it does not really minimize the variance) is compensated for by something else, presumably a slightly better (less noisy) convergence. In the same section, an alternative method was introduced which should combine convergence to minimal variance with the computational simplicity of the cross-entropy method. Given that the variance of the cross-entropy method is not higher than that of the variance-minimization method, little advantage can be expected from the alternative method. Indeed, the simulation results in the table confirm this.

The last method (the "precise" DTMC method) presented in the table is the version of the DTMC method which exactly minimizes the cross-entropy, whereas the second last method is the version that does this only approximately, but saves a lot of computational effort; see Section 7.3.1 for the details. It is clear from the results in the table that both methods perform equally well; therefore, in the rest of this section only the approximate but computationally efficient version will be used.

7.4.2 Two non-Markovian queues in tandem with feedback

To check the adaptive importance sampling method for non-Markovian systems, we use the same network as in the previous example (Figure 7.2), but with different distributions. The interarrival time distribution is set to a two-stage Erlang distribution with exponential parameter 0.2. The service time distributions are set to uniform on $[0, 3.333]$ and $[0, 5]$, for the first and the second server, respectively. Note that with these settings, the expectations of the interarrival and service times are the same as those in the previous example, so also the server utilizations are identical.

Obviously, the DTMC method is not applicable to this problem. Between the remaining three methods (variance minimization, cross-entropy, and the alternative variance-minimization), little performance difference was observed in the previous example. Because of its computational advantages, only the cross-entropy method was tried for the present model.

The results are presented in Table 7.2. Like before, 10^3 replications were simulated in every iteration, except for the last one, in which 10^5 many were used. In the table the following symbols are used: λ is the exponential parameter of the Erlang-2 interarrival time distribution; θ_1 and θ_2 are the exponential tilting parameters of the service time distributions; p is the routing probability, just like in the previous example.

The table clearly shows that also for this non-Markovian system, the simulation parameters quickly converge to their final value. Checking the correctness of

iteration	repl.	λ	θ_1	θ_2	p	estimate	rel.s.d.
1	10^3	0.2	0	0	0.5	0	∞
2	10^3	0.330	$-5.43 \cdot 10^{-2}$	0.083	0.231	$3.201 \cdot 10^{-25}$	0.1325
3	10^3	0.364	$-2.53 \cdot 10^{-10}$	0.146	0.237	$3.074 \cdot 10^{-25}$	0.0329
4	10^3	0.359	$4.96 \cdot 10^{-8}$	0.157	0.235	$3.353 \cdot 10^{-25}$	0.0367
5	10^3	0.357	$-6.67 \cdot 10^{-3}$	0.154	0.239	$3.322 \cdot 10^{-25}$	0.0313
6	10^3	0.355	$-1.57 \cdot 10^{-3}$	0.155	0.237	$3.351 \cdot 10^{-25}$	0.0421
7	10^3	0.356	$-3.26 \cdot 10^{-8}$	0.154	0.239	$3.154 \cdot 10^{-25}$	0.0341
7	10^5	0.356	$-3.26 \cdot 10^{-8}$	0.154	0.239	$3.285 \cdot 10^{-25}$	0.0036

Table 7.2: Experimental results for a non-Markovian feedback tandem queue.

the probability estimates is not possible, since no analytical results are available for this system. However, all estimates are consistent with each other (with the given standard deviations), and the relative error decreases by approximately a factor of 10 when the number of replications is increased by a factor of 100, so the results seem reliable.

7.4.3 Jackson network with two sources, routing and feedback

Consider the three-node Jackson network depicted in Figure 7.3. This network has two inputs, each of which is fed by a source with exponentially distributed inter-arrival times. Furthermore, after completing service in server 2, the customers can leave the network (with probability p_2), or enter queue 3. After completing service in server 3, the customers again can leave the system (with probability p_3), or return to queue 2. All servers have exponentially distributed service times. The rare event of interest is overflow of the total population of the network.

The following parameter values were used:

$$\lambda_1 = \lambda_2 = 1 \qquad \mu_1 = \mu_2 = \mu_3 = 6 \qquad p_2 = p_3 = 1/2.$$

Due to the feedback and the routing, the load offered to the first queue was 1/6, to the second queue 4/9, and to the third queue 2/9. Clearly, the second queue is the bottleneck. The overflow level of the total network population was set to 50.

The results of applying the variance-minimization, the cross-entropy and the DTMC method to this system are shown in Table 7.3. Initially, 10^4 replications were used for every simulation; the last few iterations (marked with a star in the table) used 10^6 replications to verify the validity of the results. The numerical method from Chapter 2 was also applied, yielding a probability of $9.8386 \cdot 10^{-18}$. Furthermore, the method from [FLA91] gives the following rates: $\lambda_1^* = \lambda_2^* = 2.25$,

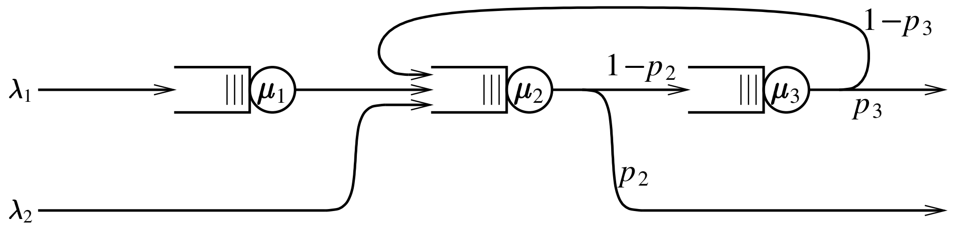


Figure 7.3: A three-node Jackson network.

$\mu_1^* = 6$, $\mu_2^* = 3.5$, $\mu_3^* = 6$, $p_2^* = 0.381$ and $p_3^* = 0.307$, which is quite close to what the adaptive methods find.

It is clear from the table that all three methods give correct estimates of the rare-event probability; in the majority of the cases the estimate is within one standard deviation from the numerical result. In spite of this, some unexpected behaviour is observed when the number of replications is increased by a factor of 100: one would expect the standard deviation to drop by a factor of $\sqrt{100} = 10$, but that only happens in the DTMC case. In the other two cases, the standard deviation drops significantly less, and increases in the next iteration. This suggests that the non-DTMC simulations are not reliable for this system.

The following experiment confirms this unreliability. About 100 iterations were performed of the cross-entropy method with 10^6 replications per iteration. Ignoring the first few iterations to allow the method to converge, almost all of the results have a relative error between 0.005 and 0.01; 14 cases had a relative error between 0.01 and 0.02, leaving only three exceptions, namely 0.082, 0.10 and even 0.31. Rerunning the exceptional iterations with a different seed for the random number generator resulted in a small relative error. These exceptions suggest that there are some sample paths that do lead to the overflow event, but are not favored by the tilting found by the adaptive procedure; when finally such an unfavored path occurs, it will have a large likelihood ratio and thus make a large contribution to the estimator and the estimated variance. This mechanism has been proposed (for other models) in [GK95] and [GW97].

7.4.4 Tandem queue with "difficult" parameter values

A well-known heuristic for determining the optimal change of measure for the simulation of overflows in tandem Jackson networks, is interchanging the arrival rate with the slowest (bottleneck) service rate; this has been suggested originally by [PW89]. In [GK95], the performance of this change of measure is studied analytically. It is shown that for some range of the arrival and service rates this method works well, leading to asymptotically efficient simulation, or even simulation with bounded relative error. However, it is also shown that for some

Variance-minimization method

iteration	λ_1	λ_2	μ_1	μ_2	p_2	μ_3	p_3	estimate	rel.s.d.
1	1	1	6	6	0.5	6	0.5	0	∞
2	1.541	1.471	5.258	4.121	0.366	5.137	0.327	$1.71 \cdot 10^{-17}$	0.5483
3	2.163	2.032	5.551	3.490	0.388	5.372	0.272	$9.04 \cdot 10^{-18}$	0.0460
4	2.169	2.080	5.839	3.539	0.377	5.705	0.307	$1.03 \cdot 10^{-17}$	0.1048
5	1.984	2.154	5.939	3.552	0.381	5.591	0.276	$9.38 \cdot 10^{-18}$	0.0464
6	2.138	2.160	5.977	3.618	0.371	5.701	0.300	$9.55 \cdot 10^{-18}$	0.0466
7	2.192	2.182	5.842	3.630	0.388	5.694	0.303	$1.08 \cdot 10^{-17}$	0.1046
8	2.246	2.137	5.518	3.691	0.348	5.306	0.327	$1.04 \cdot 10^{-17}$	0.0690
9	2.138	2.079	5.887	3.632	0.382	5.944	0.312	$1.03 \cdot 10^{-17}$	0.0613
9*	2.138	2.079	5.887	3.632	0.382	5.944	0.312	$9.78 \cdot 10^{-18}$	0.0059
10*	2.125	2.122	5.884	3.633	0.381	5.847	0.309	$9.96 \cdot 10^{-18}$	0.0118
11*	2.114	1.966	6.003	3.742	0.376	5.796	0.325	$9.80 \cdot 10^{-18}$	0.0074

Cross-entropy method

iteration	λ_1	λ_2	μ_1	μ_2	p_2	μ_3	p_3	estimate	rel.s.d.
1	1	1	6	6	0.5	6	0.5	0	∞
2	1.818	1.647	5.268	4.106	0.310	5.036	0.281	$1.28 \cdot 10^{-17}$	0.2848
3	2.025	2.152	6.307	3.442	0.358	5.843	0.302	$9.56 \cdot 10^{-18}$	0.0662
4	2.210	2.183	5.895	3.565	0.384	5.894	0.310	$9.54 \cdot 10^{-18}$	0.0378
5	2.211	2.189	5.881	3.558	0.376	5.883	0.308	$9.26 \cdot 10^{-18}$	0.0324
6	2.198	2.183	5.934	3.586	0.376	5.871	0.311	$1.01 \cdot 10^{-17}$	0.0457
7	2.182	2.160	5.928	3.585	0.376	5.909	0.305	$9.68 \cdot 10^{-18}$	0.0490
8	2.210	2.143	5.908	3.568	0.369	5.835	0.313	$1.01 \cdot 10^{-17}$	0.0419
8*	2.210	2.143	5.908	3.568	0.369	5.835	0.313	$9.72 \cdot 10^{-18}$	0.0069
9*	2.200	2.167	5.906	3.571	0.376	5.866	0.308	$9.84 \cdot 10^{-18}$	0.0150
10*	2.199	2.172	5.914	3.577	0.375	5.851	0.310	$9.97 \cdot 10^{-18}$	0.0102

DTMC method

iteration	λ_1	λ_2	μ_1	μ_2	p_2	μ_3	p_3	estimate	rel.s.d.
1	1	1	6	6	0.5	6	0.5	0	∞
2	0.124	0.104	0.284	0.221	0.356	0.266	0.316	$9.43 \cdot 10^{-18}$	0.0552
3	0.114	0.114	0.298	0.180	0.375	0.294	0.313	$9.71 \cdot 10^{-18}$	0.0132
4	0.114	0.114	0.298	0.179	0.375	0.295	0.308	$9.73 \cdot 10^{-18}$	0.0128
5	0.114	0.113	0.299	0.179	0.376	0.295	0.307	$9.62 \cdot 10^{-18}$	0.0126
6	0.114	0.113	0.299	0.179	0.377	0.295	0.307	$9.64 \cdot 10^{-18}$	0.0130
7	0.114	0.113	0.300	0.179	0.376	0.295	0.307	$9.88 \cdot 10^{-18}$	0.0123
8	0.114	0.113	0.299	0.179	0.375	0.295	0.306	$9.75 \cdot 10^{-18}$	0.0135
8*	0.114	0.113	0.299	0.179	0.375	0.295	0.306	$9.85 \cdot 10^{-18}$	0.0015
9*	0.114	0.113	0.300	0.179	0.376	0.295	0.308	$9.85 \cdot 10^{-18}$	0.0015
10*	0.114	0.113	0.299	0.179	0.376	0.295	0.308	$9.86 \cdot 10^{-18}$	0.0015

Table 7.3: Experimental results for the three-node Jackson network.

other range of the arrival and service rates, the simulation is not asymptotically efficient.

In this section, we apply the DTMC method[5] to two queues in tandem, with

[5]Only the DTMC method was used because in all of the previous experiments (except for the non-

arrival rate 0.04 and service rate at both servers of 0.48. This is in the region where according to [GK95], interchanging the arrival rate and the bottleneck service rate does not yield an asymptotically efficient simulation. The simulation results are shown in Table 7.4, in the familiar format, for three overflow levels: 12, 25 and 50. For comparison, the exact values of the overflow probabilities (calculated numerically according to Chapter 2) are also shown in the table.

For overflow levels of 12 and 25, the simulation procedure clearly yields correct estimates. However, in the case of overflow level 25, a rather large number of replications is needed for the relative error to become acceptably low (10^6 replications instead of 10^3 or 10^4 in earlier examples). For an overflow level of 50, the probability estimate turns out to be off by more than a factor of 2 in most simulations, even though the estimated relative error is small in many cases. Increasing the number of replications does not help. This is an indication that the method yields an infinite variance estimator in this case.

7.4.5 Single queue with Markov-modulated source

All of the preceding examples contained a simple source model with independent and identically distributed interarrival times. In the present example, we consider a model with a Markov-modulated source. The modulating Markov-chain is chosen such that the resulting source can also be considered as the aggregate of many independent sources which alternate between an "on" state, during which arrivals are periodic at a constant rate, and an "off" state, during which no arrivals are produced. The durations of the on and off periods are exponentially distributed with rates α and β, respectively; the resulting Markov chain is depicted in Figure 7.4.

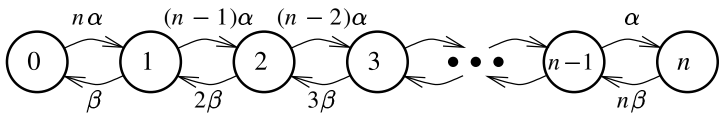


Figure 7.4: Modulating Markov chain.

Simulating one step in this modulating Markov chain consists of sampling from two random variables. The first represents the holding time at the current state; this has an exponential distribution whose rate is the total rate out of the state. The second has a binomial distribution, to decide whether to jump to the next higher or the next lower state. Obviously, the rate of the holding time distribution and the parameter of the binomial distribution are different for each

Markovian ones, of course) it performed equally well or better than the other methods. A test of the present model with the (non-DTMC) CE method and overflow level 50 showed no improvement over the DTMC method.

Overflow level = 12 (exact probability $= 1.4693 \cdot 10^{-11}$)

iteration	replications	λ	μ_1	μ_2	estimate	rel.s.d.
1	10^4	0.04	0.48	0.48	0	∞
2	10^4	0.522	0.412	0.066	$1.440 \cdot 10^{-11}$	0.0325
3	10^4	0.582	0.345	0.073	$1.520 \cdot 10^{-11}$	0.0262
4	10^4	0.577	0.347	0.076	$1.462 \cdot 10^{-11}$	0.0248
4	10^5	0.577	0.347	0.076	$1.455 \cdot 10^{-11}$	0.0104
5	10^5	0.576	0.346	0.078	$1.458 \cdot 10^{-11}$	0.0082
6	10^5	0.577	0.347	0.076	$1.470 \cdot 10^{-11}$	0.0084
6	10^6	0.577	0.347	0.076	$1.472 \cdot 10^{-11}$	0.0026

Overflow level = 25 (exact probability $= 2.8722 \cdot 10^{-25}$)

iteration	replications	λ	μ_1	μ_2	estimate	rel.s.d.
1	10^4	0.04	0.48	0.48	0	∞
2	10^4	0.522	0.412	0.066	$2.258 \cdot 10^{-25}$	0.1301
3	10^4	0.565	0.379	0.056	$2.537 \cdot 10^{-25}$	0.1654
4	10^4	0.592	0.339	0.068	$3.806 \cdot 10^{-25}$	0.2622
5	10^4	0.627	0.290	0.083	$4.610 \cdot 10^{-25}$	0.3314
6	10^4	0.566	0.382	0.052	$2.300 \cdot 10^{-25}$	0.2370
6	10^5	0.566	0.382	0.052	$2.978 \cdot 10^{-25}$	0.1283
7	10^5	0.582	0.342	0.076	$3.142 \cdot 10^{-25}$	0.0609
8	10^5	0.590	0.335	0.074	$2.933 \cdot 10^{-25}$	0.0762
9	10^5	0.576	0.342	0.082	$2.787 \cdot 10^{-25}$	0.0762
9	10^6	0.576	0.342	0.082	$3.054 \cdot 10^{-25}$	0.0354
10	10^6	0.581	0.343	0.076	$2.882 \cdot 10^{-25}$	0.0197
11	10^6	0.585	0.340	0.074	$2.869 \cdot 10^{-25}$	0.0279

Overflow level = 50 (exact probability $= 6.0327 \cdot 10^{-52}$)

iteration	replications	λ	μ_1	μ_2	estimate	rel.s.d.
1	10^4	0.04	0.48	0.48	0	∞
2	10^4	0.522	0.412	0.066	$3.181 \cdot 10^{-52}$	0.4357
3	10^4	0.516	0.435	0.049	$3.598 \cdot 10^{-52}$	0.4430
4	10^4	0.558	0.390	0.051	$2.791 \cdot 10^{-51}$	0.8905
5	10^4	0.572	0.378	0.050	$3.402 \cdot 10^{-52}$	0.5086
5	10^5	0.572	0.378	0.050	$2.414 \cdot 10^{-52}$	0.1280
6	10^5	0.529	0.424	0.047	$3.288 \cdot 10^{-52}$	0.2883
7	10^5	0.516	0.425	0.059	$2.081 \cdot 10^{-52}$	0.0622
8	10^5	0.522	0.430	0.048	$2.473 \cdot 10^{-52}$	0.1076
8	10^6	0.522	0.430	0.048	$2.712 \cdot 10^{-52}$	0.0512
9	10^6	0.531	0.418	0.051	$4.160 \cdot 10^{-52}$	0.2527
10	10^6	0.563	0.386	0.051	$3.413 \cdot 10^{-52}$	0.0781
11	10^6	0.539	0.404	0.056	$2.988 \cdot 10^{-52}$	0.0538
12	10^6	0.536	0.412	0.053	$2.905 \cdot 10^{-52}$	0.0509
13	10^6	0.535	0.413	0.052	$4.483 \cdot 10^{-52}$	0.1989
13	10^7	0.535	0.413	0.052	$4.343 \cdot 10^{-52}$	0.0852
14	10^7	0.546	0.397	0.057	$4.700 \cdot 10^{-52}$	0.1559
14	10^8	0.546	0.397	0.057	$11.613 \cdot 10^{-52}$	0.5837

Table 7.4: Experimental results for two queues in tandem in the "difficult" parameter region.

state of our Markov chain. As a consequence, the simple cross-entropy formulas from Section 7.2.2 cannot be applied, so one needs to go back to the basic cross-entropy minimization equations (7.4) and (7.5).

A natural choice for the tilting is to simply change the rates α and β. The question then is how to change α and β such that the maximum in (7.4) (or (7.5)) is attained. By performing an analysis similar to Section 7.2.2 (albeit a bit more complicated), one finds that

$$\alpha^{\dagger} = a/b \qquad \text{and} \qquad \beta^{\dagger} = (1-a)/c,$$

where a, b and c are averages of three quantities over all samples on paths on which the rare event is reached; a is such an average of the samples of the binomial random variable with 0 indicating a step to the next lower state and 1 a step to the next higher state; b is such an average of $x \cdot (n-i)$, where x is the sample from the exponential distribution and i is the state of the Markov chain; finally, c is such an average of $x \cdot i$. Note that the right-hand sides of (7.7) and (7.8) are also such averages.

iteration	replications	α	β	estimate	rel.std.dev.
1	10^4	0.005	0.020	0	∞
2	10^4	0.0710	0.0300	$6.049 \cdot 10^{-16}$	0.4384
3	10^4	0.0503	0.0403	$3.515 \cdot 10^{-15}$	0.2277
4	10^4	0.0423	0.0484	$4.727 \cdot 10^{-15}$	0.0545
5	10^4	0.0420	0.0509	$4.412 \cdot 10^{-15}$	0.0514
6	10^4	0.0417	0.0509	$4.334 \cdot 10^{-15}$	0.0543
6	10^6	0.0417	0.0509	$4.401 \cdot 10^{-15}$	0.0053

Table 7.5: Experimental results for a single queue with a Markov-modulated source.

Table 7.5 shows the simulation results for the buffer overflow probability of a single queue with the source model described above, and a deterministic service time. The parameters are as follows: service time = 1/3; interarrival time = $1/i$, where i is the state of the modulating Markov process; $\alpha = 0.02$, $\beta = 0.1$ and $n = 10$. The resulting modulated arrival process corresponds to 10 on/off sources, each transmitting (when on) at 1/3 of the service rate, with an average burst size of 10, and average silence time of 50 time units. We estimate the probability that an overflow of the queue occurs before the end of a busy period, starting at the first arrival of the busy period and with the modulating Markov chain in state 2. The overflow level was set to 400. Judging from the simulation results in the table, the method works well in this example.

In Chapter 3 of [Man96], importance sampling simulation of a Markov-modulated *fluid* model is discussed. Applying those calculations to a fluid model with the parameter values used in the above discrete-arrival model, gives $\alpha = 0.0428$ and $\beta = 0.0467$; these are close to the simulation results obtained above for the model with discrete arrivals.

7.4.6 Asymptotic efficiency

To check the asymptotic efficiency, several of the previous experiments have been repeated with different overflow levels. Table 7.6 shows the results. Since in Sections 7.4.1 through 7.4.3 convergence was always achieved by the fourth iteration, in principle only the results from the fourth iteration are shown in the table; however, there are a few exceptions, marked with a star.

system	method	level	exact	estimate	rel.std.dev.
Markov tandem queue with feedback (10^3 repl.)	cross-entropy	25	$8.940 \cdot 10^{-8}$	$8.780 \cdot 10^{-8}$	0.0380
		50	$2.665 \cdot 10^{-15}$	$2.408 \cdot 10^{-15}$	0.0436
		100	$2.367 \cdot 10^{-30}$	$2.353 \cdot 10^{-30}$	0.0411
		200	$1.867 \cdot 10^{-60}$	$1.597 \cdot 10^{-60}$	0.0420
	DTMC	25	$8.940 \cdot 10^{-8}$	$9.145 \cdot 10^{-8}$	0.0367
		50	$2.665 \cdot 10^{-15}$	$2.805 \cdot 10^{-15}$	0.0364
		100	$2.367 \cdot 10^{-30}$	$2.341 \cdot 10^{-30}$	0.0391
		200	$1.867 \cdot 10^{-60}$	$1.874 \cdot 10^{-60}$	0.0413
Non-Markov tandem queue with feedback (10^3 repl.)	cross-entropy	25	–	$1.596 \cdot 10^{-12}$	0.0365
		50	–	$3.353 \cdot 10^{-25}$	0.0367
		100	–	$1.489 \cdot 10^{-50}$	0.0288
		200	–	$3.216 \cdot 10^{-101}$	0.0385
Three-node Jackson network (10^4 repl.)	cross-entropy			see text	
	DTMC	25	$6.273 \cdot 10^{-9}$	$6.296 \cdot 10^{-9}$	0.0121
		50	$9.839 \cdot 10^{-18}$	$9.731 \cdot 10^{-18}$	0.0128
		100	–	$2.433 \cdot 10^{-35}$	0.0139
		200	–	$1.454 \cdot 10^{-70}$	0.0141*
Two node tandem (10^6 repl.)	DTMC	12	$1.469 \cdot 10^{-11}$	$1.472 \cdot 10^{-11}$	0.0026**
		25	$2.872 \cdot 10^{-25}$	$2.869 \cdot 10^{-25}$	0.0279**
		50	$6.033 \cdot 10^{-52}$	incorrect estimate	
Single queue with Markov-modulated source (10^4 repl.)	cross-entropy	100	–	$6.374 \cdot 10^{-6}$	0.0542
		200	–	$5.964 \cdot 10^{-9}$	0.0573
		400	–	$4.727 \cdot 10^{-15}$	0.0545
		800	–	$2.871 \cdot 10^{-27}$	0.0526*

* fifth instead of fourth iteration because of significantly better relative error

** later iteration, see Table 7.4

Table 7.6: Test of the asymptotic efficiency.

It is clear from the table that in most cases the relative error grows hardly or not at all with the overflow level (and thus the rarity of the event). Thus, we conclude that in those cases the method is asymptotically efficient, and even has a bounded relative error.

No results have been given for the three-node Jackson network with cross-entropy simulation. This is because of the varying relative error, already noted in Section 7.4.3. In the course of iterations, these relative errors tend to vary about a factor of 2, with every now and then a much bigger value. Still, it can be noted that this factor 2 range is roughly the same for all overflow levels tried. But it does not seem warranted to conclude asymptotic efficiency from this, as long as the meaning of the observed exceptionally large relative errors is still unclear.

Another exception is the tandem queue with the specific parameter setting studied in Section 7.4.4. In this case, the relative error increases by a factor of 10 when (approximately) doubling the overflow level from 12 to 25; after doubling the overflow level again, the simulation gives incorrect estimates, even for a higher number of replications. Clearly, this is not an asymptotically efficient simulation.

7.4.7 Other rare events

In the above, only overflows of the total network population during a busy cycle have been considered. However, the method is not limited to such problems. Other problems for which the method is applicable include overflow of one particular queue in a network, and different kinds of initial and absorbing states, like the pseudo-regenerations studied in [KN99].

Applying the method to such problems yields a similar picture to the examples studied here: in some cases it works fine, in others it shows similar problems (like bad convergence and relative error not decreasing properly).

7.5 Conclusions

In this chapter, three variants of an adaptive importance sampling method have been discussed. These methods attempt to iteratively find the optimal set of simulation parameters. They differ in the algorithm used to choose the simulation parameters for the next iteration.

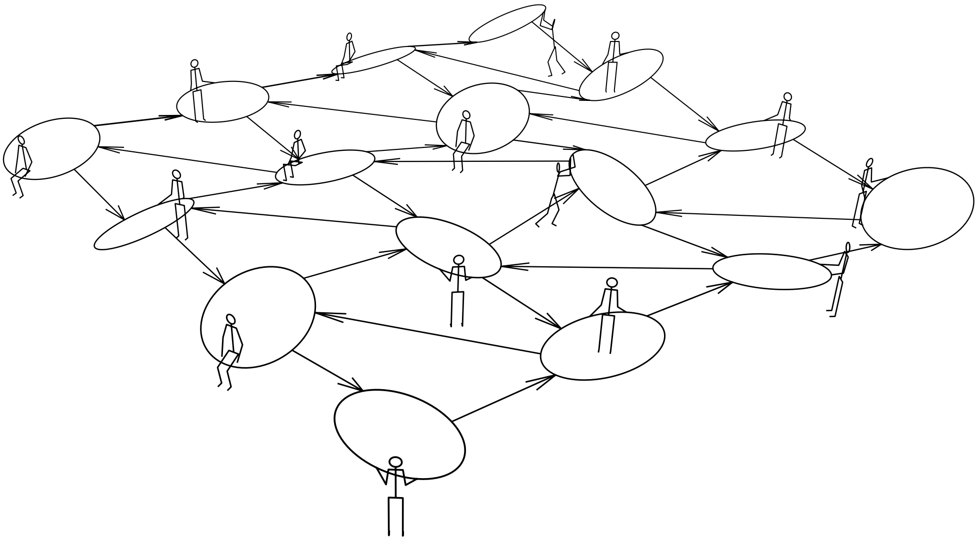
In several experiments, it was found that the methods often work well, yielding correct estimates of the rare-event probability of interest, and typically displaying asymptotic efficiency, often even bounded relative error.

However, in some cases (only at extremely low probabilities and/or at rather specific parameter settings) the following types of "bad" behaviour were noted:

(a) relative error not decreasing properly with increasing number of replications; (b) convergence irregularity (in Table 7.3); (c) completely wrong estimate (in Table 7.4). Clearly, the existance of such problems (especially the third one; in the first two cases the resulting estimates seem to be correct) means that the method is not generally applicable and fails in some cases.

A possible explanation for these problems is that the tilting found by the adaptive algorithm does not favor all typical paths leading to the rare event well enough. Thus, the contribution from an unfavored but typical path will be zero in most replications, but large in some replications, causing a lot of variance, and irregular results if the number of replications is too small to sufficiently sample the unfavored typical path. This problem has been discussed in the literature, in [GK95] and [GW97]. In the latter paper, a possible remedy is proposed: split the rare event into several separate events, such that each of them can be simulated well, and then combine the results of the separate simulations; some simple (non-queueing) examples are provided, but it is not clear how to apply this idea in a queueing context.

In the next chapter, we propose a variant of the cross-entropy method from the present chapter. In this variant, we allow the change of measure (tilting) to depend on the state of the system; the cross-entropy technique is still used to adaptively find the optimal such tilting. As will be shown, this method yields asymptotically efficient estimation of overflow probabilities such as those of the tandem queue in Section 7.4.4, where the present chapter's method (and large-deviations heuristics) fail.



Chapter 8

Adaptive importance sampling simulation with state-dependent tilting

In the previous chapter, several methods have been proposed for adaptively choosing the (tilting) parameters in an importance sampling simulation. All of them share an important characteristic: the tilting does not depend on the state of the system. It was shown that these methods work quite well for some systems, but not for others; typical problems include unreliable simulation, convergence problems, and a growing relative error (the simulation is possibly no longer asymptotically efficient). In fact, there is evidence in the literature (e.g. [GK95]) that such a state-indepent tilting simply cannot result in asymptotically efficient simulation for some problems, like the tandem queue with certain parameter values.

In this chapter, an alternative method will be explored in which the tilting parameters are allowed to depend on the state of the system. We realize this by assigning a separate set of tilting parameters to every distinct state of the system, in order to not unnecessarily restrict the way the tilting depends on the state. Clearly, this gives an immense freedom to the tilting, so improved performance of the simulation is to be expected. On the other hand, it makes the problem of correctly choosing the tilting parameters much harder: a few parameters need to be chosen *per state*, as opposed to only a few parameters *in total* in the state-independent case.

The idea of allowing the tilting to depend on the state of the systems seems to have received relatively little attention in the literature. In [CFM83], an efficient exponential change of measure is considered for a certain class of Markov chain problems, where the exponential tilting parameter is a function of the state and is determined using large-deviations theory. In [Hee98b], a DTMC simulation of the load on a network is described, where the transition probabilities at each state of the DTMC are chosen on a heuristic basis. Furthermore, in [KN99] a DTMC model of a two-node tandem network is simulated with a tilting that depends on the contents of the first buffer; the details of this tilting are based on modelling the system as a Markov additive process. In [MR00], transient overflows of a queue with a large number of Markov-modulated fluid sources are simulated with a change of measure depending on the time. In the latter methods, the tilting is based on a mathematical model, which makes the results rather specific to these particular problems. In the method to be discussed in this chapter, the tilting is not determined in advance; instead, it will be chosen adaptively using an iterative procedure based on cross-entropy, similar to the method discussed in the previous chapter.

In Section 8.1, the method will be described in detail. As noted above, the biggest problem with the method is the enormous number of tilting parameters for models with a large state-space; Section 8.2 discusses ways to overcome this. The performance of the method is studied experimentally in Section 8.3, followed by a mathematical analysis of some aspects of the method in Section 8.4. All of these sections only discuss DTMC simulations. Section 8.5 provides some concluding remarks, including a brief discussion of the possiblity of extending the method to non-Markovian problems.

8.1 Principles

In principle, extending the state-independent methods from Chapter 7 to a state-dependent method is trivial: the system to be simulated is modified such that each of its random variables (like interarrival and service times) is replaced by a set of identically distributed random variables, one for each of the system's states. Correspondingly, the simulation procedure is modified such that whenever it needs to sample a random variable, it samples the copy of that random variable belonging to the current state of the system. Then each of the copies of the distribution can be assigned its own, separate, tilting factor: the tilting can be made state-dependent. The rest of the method as discussed in Chapter 7 still applies.

In fact, things are not quite as simple as this. For non-DTMC systems, the state space generally is not discrete, so it is not possible to assign a different set of tilting parameters to every state; in fact there are more problems with non-DTMC systems, see Section 8.5.2. For DTMC models, direct application of the method from Section 7.3 to the system modified as above would involve inverting a very large matrix in (7.14). The matrix would have a structure which could be exploited to make the inversion feasible, but it is simpler to derive the equations for state-dependent DTMC simulation from the basic cross-entropy minimization equation (7.4), as we will do below.

8.1.1 Preliminaries

As stated before, we will restrict the derivations to DTMC models. Such models are completely described by their initial probability distribution and their set of transition probabilities, i.e., the probabilities of going from one state to another. Since many DTMC models (e.g., for queueing systems) are derived from continuous time models with exponential time distributions (CTMCs), the transition probabilities are typically calculated from transition rates: the probability of going from state i to state j is given by $\lambda_{ij}/\sum_k \lambda_{ik}$, where λ_{ij} is the transition rate from state i to state j, and k in $\sum_k$ runs over all states. In fact, many of the calculations done in this chapter can most conveniently be performed in terms of rates instead of transition probabilities, because this avoids the need to continuously take the condition that the probabilities must sum up to 1 into account. Therefore, all calculations for establishing the optimal tilting will be done in terms of the rates; whenever real transition probabilities are needed (e.g., to actually perform the simulation), they can be trivially calculated by normalizing the sum of all rates out of a state to 1.

In DTMC models, only one type of tilting is possible: changing the transition probabilities. Thus, the tilting can be specified by giving the new set of transition probabilities. However, as noted above, it is most convenient to work in terms of rates, so we will actually specify the tilting as a vector $\boldsymbol{\lambda}$ of all (tilted) transition rates λ_{ij}. The aim then is to find a $\boldsymbol{\lambda}$ which minimizes the variance of the importance sampling estimator.

Before deriving the actual cross-entropy and variance minimization formulas, let us first build a mathematical description of one replication Z of a DTMC simulation; note that this description is different from the one in Section 7.2.1, because different details need to be emphasized. Define the sequence z_i which denotes the state of the system just before the ith transition in this replication Z. Denote by λ_{lm} the rate (or probability) of going from state l to state m. Then obviously the (a priori) probability of the ith step is

$$\frac{\lambda_{z_i z_{i+1}}}{\sum_k \lambda_{z_i k}},$$

where k runs over all states (or, equivalently, only those states that can be reached in one step from state z_i, since all other $\lambda_{z_i k}$ are 0). The total probability of the sample path Z is

$$\mathbb{P}(Z) = \prod_i \frac{\lambda_{z_i z_{i+1}}}{\sum_k \lambda_{z_i k}},$$

where i runs over all steps in the sample path.

8.1.2 Cross-entropy formulation

Substitute the above expression for the probability of a sample path into (7.4); then we get the following expression for the optimal transition rate vector $\boldsymbol{\lambda}^\dagger$:

$$\boldsymbol{\lambda}^\dagger = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_0 I(Z) \ln \prod_i \frac{\lambda_{z_i z_{i+1}}}{\sum_k \lambda_{z_i k}} = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_0 I(Z) \sum_i \left(\ln \lambda_{z_i z_{i+1}} - \ln \sum_k \lambda_{z_i k} \right).$$

To find the maximum in the right-hand side, set the derivative with respect to λ_{lm} to 0, for any two states l and m:

$$0 = \mathbb{E}_0 I(Z) \sum_{i:z_i=l} \left(\frac{1_{(z_{i+1}=m)}}{\lambda^\dagger_{lm}} - \frac{1}{\sum_k \lambda^\dagger_{lk}} \right),$$

or, equivalently:

$$\frac{1}{\lambda^\dagger_{lm}} \mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1_{(z_{i+1}=m)} = \frac{1}{\sum_k \lambda^\dagger_{lk}} \mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1.$$

Thus, we find the following expression for the optimal transition probability q_{lm} from state l to state m:

$$q_{lm} = \frac{\lambda^\dagger_{lm}}{\sum_k \lambda^\dagger_{lk}} = \frac{\mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1_{(z_{i+1}=m)}}{\mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1}. \tag{8.1}$$

Of course, the expectations in the right-hand side are generally not known, but we can approximate them as follows:

$$\begin{aligned} q_{lm} = \frac{\lambda^\dagger_{lm}}{\sum_k \lambda^\dagger_{lk}} &= \frac{\mathbb{E}_{\boldsymbol{\lambda}_j} I(Z) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1_{(z_{i+1}=m)}}{\mathbb{E}_{\boldsymbol{\lambda}_j} I(Z) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1} \\ &\approx \frac{\sum_{Z=Z_1}^{Z_N} I(Z) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1_{(z_{i+1}=m)}}{\sum_{Z=Z_1}^{Z_N} I(Z) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1}, \end{aligned} \tag{8.2}$$

where $\sum_{Z=Z_1}^{Z_N}$ is a sum over the sample paths from N replications, simulated with transition rates $\boldsymbol{\lambda}_j$ (see below). Note that the factor $\sum_{i:z_i=l} 1$ in the denominator is just the number of visits to state l during replication Z, and that $\sum_{i:z_i=l} 1_{(z_{i+1}=m)}$

in the numerator is the number of those visits in which the transition to state m was chosen next.

The j in $\boldsymbol{\lambda}_j$ in the above typically refers to the iteration number, as it did in Chapter 7. Thus, given an initial set of transition rates $\boldsymbol{\lambda}_1$, we can perform a simulation and use (8.2) to find the second set of transition rates $\boldsymbol{\lambda}_2$; actually, we can only find these rates up to a constant factor, but this is enough for a DTMC simulation. Next, a simulation using $\boldsymbol{\lambda}_2$ can be used to find $\boldsymbol{\lambda}_3$, and so on. Basically, the algorithm from Section 7.1.4 can be applied, using (8.2) in step 4 to calculate the tilting $\boldsymbol{\lambda}_{j+1}$ for the next iteration.

8.1.3 Variance-minimization formulation

A derivation similar to the above can be started from equation 7.2, yielding

$$\frac{\lambda^*_{lm}}{\sum_k \lambda^*_{lk}} = \frac{\mathbb{E}_{\boldsymbol{\lambda}_j} I(Z) L(Z, \boldsymbol{\lambda}) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1_{(z_{i+1}=m)}}{\mathbb{E}_{\boldsymbol{\lambda}_j} I(Z) L(Z, \boldsymbol{\lambda}) L(Z, \boldsymbol{\lambda}_j) \sum_{i:z_i=l} 1}.$$

Compared to the cross-entropy version (8.2), an additional likelihood ratio factor has appeared. This difference between the cross-entropy and the variance-minimization formulation is not surprising: it was also observed in Section 7.2.3.

As in the case of state-independent tilting, the cross-entropy formulation has computational advantages over the variance-minimization approach, so only the former will be used in this chapter.

8.1.4 Practical problems

Using the adaptive importance sampling method with state-dependent parameters chosen according to (8.2) seems very simple. There are, however, practical difficulties. The cause of these is the enormous number of states that a typical queueing network can have. For example, a network with three queues and an overflow level of 50 for the total network population (like the network studied in Section 7.4.3) has 23425 states[1]. Doubling the overflow level to 100 multiplies this number of states by almost 8. If the rare event of interest is the overflow of one particular queue, other queues in the network can have an infinite size, thus making the number of states infinite.

One of the consequences of the enormous state space is that a lot of data needs to be stored: this takes a lot of memory capacity; but with present-day computers and the size of the queueing networks studied here, this is typically not a problem (except if the state space is infinite, of course). However, manipulating such a lot

[1] This is the total number of ways to distribute among three distinct queues a total of 1 customer (3 ways), 2 indistinguishable customers (6 ways), 3 indistinguishable customers (10 ways), up to 50 indistinguishable customers.

of data (e.g. in the smoothing techniques that will be discussed later) can be prohibitively time-consuming.

The accuracy of the estimations in the right-hand side of (8.2) is more problematic. The only sample paths that give a contribution to the sums in the numerator and denominator are those that both reach the rare event (because of the $I(Z)$ factor) and pass through the state l (because of the summation over i for which $z_i = l$). The factor $I(Z)$ will not be a problem: either the tilting in the jth iteration is such that the event of interest is no longer rare, or the event is modified such that it is not too rare (cf. Section 7.1.3). However, the tilting will not favor visits to states that are away from some optimal path to the rare event of interest. If the state space is multi-dimensional, this means that many states will still not be visited often or at all, even under a tilting that makes the target event non-rare. States that are not visited at all during the N replications of a simulation yield 0/0 (undefined) in the right hand side of (8.2). And states that are visited only a few times make the quotient of sums in the right-hand side a bad approximation of the quotient of expectations.

There is in fact a rather fundamental risk here: suppose the transition from some state l to another state m happens in only 10 % of all visits to state l, and state l is visited only 5 times during the N replications of a simulation. Then it is quite likely that in none of those 5 visits to state l, a transition to state m will be made. Consequently, using (8.2) to choose the simulation parameters for the next iteration would set the rate (probability) of this transition to 0, thus making the transition impossible. Then in the next simulation, surely no transitions from state l to state m will be observed, so this rate will again be set to 0 for the next iteration: it will remain at 0 forever, thus possibly resulting in a biased estimator.

The only case in which the above does not give a biased estimator, is when the rare event of interest can no longer be reached after that particular transition has been made. As a matter of fact, all paths Z which contain such a transition necessarily have $I(Z) = 0$; as a consequence, (8.2) will automatically set the rate of such a transition to zero for the next iteration. Therefore, after the first iteration, *all* sample paths will reach the rare event.

8.2 Dealing with the large number of states

As discussed above, the large state space of typical (queueing) models poses some problems for the adaptive optimization of the state-dependent change of measure. In this section, techniques are discussed to deal with these problems. These techniques exploit the highly regular structure of DTMCs corresponding to practical (queueing) models: in such DTMCs, many states are "similar" to other states, in terms of their position in the state space and their transition probabilities to their neighbour states. It is reasonable to assume that the optimal importance sampling transition probabilities for two such similar states are close. If this is the case, the estimates of the transition probabilities for a given state may be improved by also including observations from sample paths passing through an appropriate set of such similar states. Of course, this introduces an error, since the optimal probabilities are generally not really equal. On the other hand, since more samples are used, the variance of the estimation decreases. Furthermore, treating several states as if they were one state saves memory for storing the transition probabilities. This is necessary for systems with an infinite number of states.

Note that the "error" discussed above does not imply that the resulting estimate of the rare-event probability will be biased; in principle that estimate will be unbiased as long as the correct likelihood ratios are used. The error in the transition probabilities only causes the variance of the resulting estimator to be larger than optimal. In fact, such errors and the associated non-optimal variance are always present, even if no grouping of states is used, due to the fact that the transition probabilities are estimated by simulation and thus subject to statistical errors.

As in Section 7.3, we will focus on DTMC models of queueing systems. The typical properties of such a DTMC were already described in Section 7.3. Briefly, they are (for a system with n queues):

- States can be labeled by n integers, each denoting the content of one buffer, and conveniently arranged at the points of an n-dimensional grid.

- Transitions typically increment one coordinate and/or decrement another coordinate by 1.

Roughly defining "similarity" of states in such a system is not hard: if the coordinates in the n-dimensional space differ little, then the states are near each other.

Three techniques for dealing with the large number of states are described in the rest of this section:

- *Local average*: if the estimate of the transition probabilities in a state is not good enough, collect data from nearby states and try again; if necessary, add data from some more states, etc.

- *Boundary layers*: group all states in which the content of a queue is large; thus, the transition probabilities are allowed to depend on that queue's content only in states where that queue is nearly empty, i.e., near a boundary of the state space.

- *Smoothing using spline fitting*: fit a smooth function (e.g., a cubic spline) through the transition probability estimates.

Mathematical formulation of grouping of states

Combining the observations from "similar" states is actually quite simple. To do this, extend the sums in the right-hand side of (8.2) as follows:

$$q_{lm} = \frac{\lambda^{\dagger}_{lm}}{\sum_k \lambda^{\dagger}_{lk}} \approx \frac{\sum_{l'} \sum_{Z=Z_1}^{Z_N} I(Z)L(Z,\boldsymbol{\lambda}_j) \sum_{i:z_i=l'} 1_{(z_{i+1}=m')}}{\sum_{l'} \sum_{Z=Z_1}^{Z_N} I(Z)L(Z,\boldsymbol{\lambda}_j) \sum_{i:z_i=l'} 1}. \tag{8.3}$$

Here l' runs over all states of which the observations should be used for the estimation of the transition probabilities for state l. State m' is the state whose position relative to state l' is the same as the position of state m relative to state l; e.g., if state m has 1 more customer in queue 2 than state l has, then state m' must also have 1 more customer in queue 2 than state l' has. Note that for all m and all l', a suitable m' must exist, otherwise the states are not similar enough to apply this technique. Practically speaking, this means that only states with the same set of enabled transitions can be combined (e.g., states in which the same queues are empty).

In fact, also states with different sets of enabled transitions could be combined, but that would require using the more complicated formulation from Section 7.3.

8.2.1 Local average

The principle of the local average method is as follows. In order to find the new transition probabilities in state l, first treat it as a separate state, and calculate the transition probabilities according to (8.2); also calculate some measure (to be defined below) for the accuracy of these transition probabilities. Check whether the accuracy is satisfactory: if so, then use the transition probabilities just calculated. Otherwise, combine the observations of this state l with the observations from some states surrounding it, and calculate the new transition probabilities using (8.3); check the accuracy again: if satisfactory, use these transition probabilities; if not, repeat the previous steps with a larger set of surrounding states, etc.

Accuracy of transition probability estimates

In order to apply the above procedure, a measure for the accuracy of the new transition probabilities needs to be defined. A first criterion obviously is the number of times the state l (and its surrounding states) has been visited: if this number is small, then the estimated probabilities are not reliable.

Second, a check can be made that no transition probabilities that should be non-zero, have become zero, as discussed in Section 8.1.4. If a transition to a given state would get a zero probability on the basis of the observations, it must be checked whether the rare (target) state can be reached from this state. If so, then it is not acceptable that the transition probability is set to zero. This reachability test is trivial in the queueing examples discussed later in this chapter, but in other problems this test may be non-trivial.

Third, we can estimate the relative error of the probabilities estimated according to (8.2). In order to do so, first rewrite this equation in terms of a real sample average, as follows:

$$q_{lm} = \frac{\lambda^\dagger_{lm}}{\sum_k \lambda^\dagger_{lk}} \approx \frac{Q_{lm}}{\sum_k Q_{lk}} \qquad \text{with} \qquad Q_{lm} = \frac{\sum_{Z=Z_1}^{Z_N} I(Z) \sum_{i:z_i=l} L(Z,\boldsymbol{\lambda}_j) 1_{(z_{i+1}=m)}}{\sum_{Z=Z_1}^{Z_N} I(Z) \sum_{i:z_i=l} 1}.$$

Thus, Q_{lm} is the average of $L(Z,\boldsymbol{\lambda}_j)1_{(z_{i+1}=m)}$ over all visits z_i to state l on sample paths that eventually reach the rare event (i.e., for which $I(Z) = 1$). Define M to be the total number of such visits, i.e., $M = \sum_{Z=Z_1}^{Z_N} I(Z) \sum_{i:z_i=l} 1$. Then

$$Q_{lm} = \frac{1}{M} \sum_{Z=Z_1}^{Z_N} \sum_{i:z_i=l} I(Z) L(Z,\boldsymbol{\lambda}_j) 1_{(z_{i+1}=m)}$$

and the variance of this estimate obviously is

$$\frac{1}{M-1}\left(\frac{1}{M} \sum_{Z=Z_1}^{Z_N} \sum_{i:z_i=l} I(Z) L^2(Z,\boldsymbol{\lambda}_j) 1_{(z_{i+1}=m)} - Q_{lm}^2\right).$$

Consequently, the relative error is (approximating $M-1$ by M):

$$\sqrt{\frac{\sum_{Z=Z_1}^{Z_N} \sum_{i:z_i=l} I(Z) L^2(Z,\boldsymbol{\lambda}_j) 1_{(z_{i+1}=m)}}{\left(\sum_{Z=Z_1}^{Z_N} \sum_{i:z_i=l} I(Z) L(Z,\boldsymbol{\lambda}_j) 1_{(z_{i+1}=m)}\right)^2} - \frac{1}{M}}. \tag{8.4}$$

Of course, this derivation and the above equation can be extended to grouping of states, similar to the derivation of (8.3).

Which states are near?

When the data available for one state is not enough to estimate the new transition probabilities with sufficient accuracy, data from nearby states needs to be collected. Since we are focussing on queueing systems where the state is determined by the numbers of customers in the queues, it makes sense to define states as near when they have approximately the same number of customers in the same queues. More precisely, we define the distance between two states as the maximum absolute difference of the numbers of customers in the queues. For example, in a system with three queues, the distance from the state in which the three queues contain 4, 5 and 6 customers, respectively, to the state in which they contain 2, 5 and 7 customers, is $\max\big(|4-2|, |5-5|, |6-7|\big) = 2$.

However, care must be taken not to group states which have a different set of enabled transitions; typically, this means that states in which a certain queue is empty (and thus the transition corresponding to a departure from that queue is impossible) can not be combined with those in which that same queue is not empty, and vice versa. This is illustrated in Figure 8.1. Starting from two particular (encircled) states in the state space of a two-node queueing system, the figure shows which states are subsequently included when allowing a progressively larger distance d.

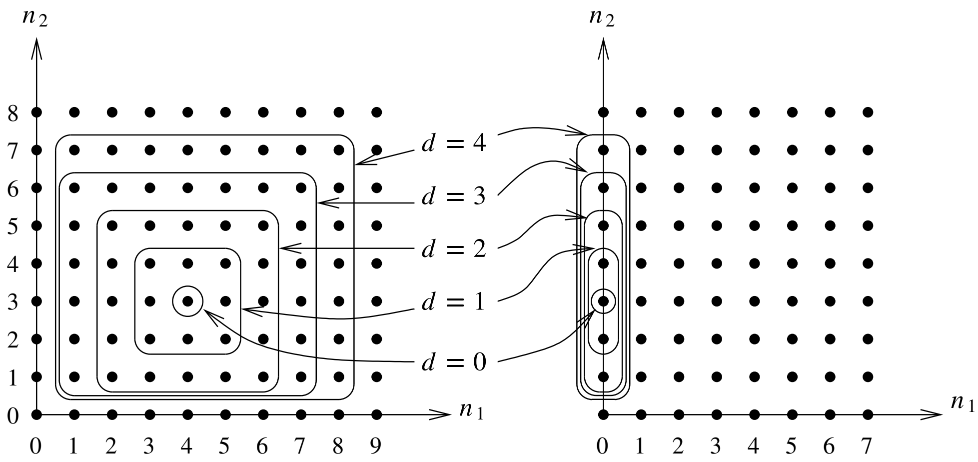


Figure 8.1: Grouping of states in the state space of a two-node queueing system.

Note that the distance as defined above is not the usual geometric distance ($\sqrt{\sum_i (x_i - y_i)^2}$, where x_i and y_i are the ith coordinates of the points x and y). One advantage of the distance defined here, is that it is very easy to loop over all points included within a certain distance from a given point; this is important, since this needs to be done possibly several times for every point, see the algorithm below. Since this works well in practice, no other distance measures were tried.

The algorithm

The above leads to the following algorithm, which should be performed for all states:

1. Initialize $d := 0$.

2. Consider all acceptable (see above) states within a distance d from the state under consideration; count the total number of visits n to these states; calculate the new transition probabilities using (8.3) and their relative errors using (8.4).

3. Check whether the number of visits n is high enough, no non-zero rates have become zero, and the highest of the relative errors is low enough. If so, stop.

4. Check whether incrementing d still enlarges the set of states. If not, stop.

5. Increment d by one, and repeat steps 2–5.

Note that for every state, a separate value of d is found, which is a measure for the locality of the resulting estimate of the transition probabilities for that state: large d means many relatively remote states have been aggregated. In the experiments section, the average value of d over all states will be shown, to give an indication of how much grouping was typically needed. The smaller the average d, the more locally the new transition probabilities were estimated.

Practical considerations

The above description still leaves a few issues open. First of all, the minimum acceptable number of visits to the (set of) states needs to be chosen. This is not really important[2], it mainly serves to ensure that all transitions have had a reasonable chance of occurring. In practice, a value of 100 was used with good results.

Secondly, the maximal acceptable relative error needs to be chosen. This is not so easy. The lower this value is chosen, the more the algorithm will be inclined to group states. Doing so reduces the variance of the transition probability estimates, but makes them less state-dependent, thus possibly worsening the estimate of the rare-event probability of interest. There is also a direct relationship with the number of replications used in the simulation: the more replications, the lower the relative error will be for a given d, so the fewer states will be grouped. In practice, setting the allowed relative error to 0.2 and then increasing the number of replications until the system converges, turned out to work well.

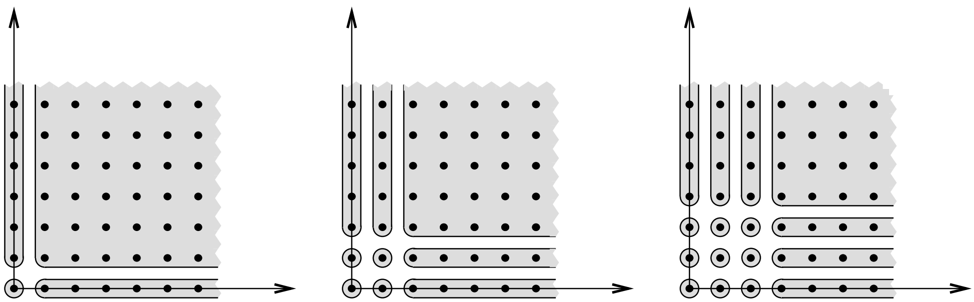
States corresponding to high levels of queues are typically reached rarely: the tilting typically only favors overflow of one queue, while for the other queues

[2] Actually, it is a relic from an earlier version of the algorithm, in which this was the only criterion. Since the test that checks whether all possible transitions have been observed has been added, one could consider dropping the criterion regarding the number of visits.

high levels remain rare. Thus, there will be many rarely visited states, for each of which the above algorithm would combine observations from many surrounding states (large d), which would be very time-consuming. Here the boundary-layer method, described in the next section, comes in: it basically combines the observations from all those high-level states in advance.

8.2.2 Boundary layers

The fundamental assumption for the boundary-layer method is that when a queue's content becomes large, the transition rates will hardly depend on that queue's content; this assumption is based on observations (see Figures 8.8 and 8.10, and Appendix 8.A). So instead of distinguishing between all possible values of the queue's content ($0, 1, 2, \ldots, K$, where K is the highest possible level, e.g., the overflow level or the maximum network population), we only distinguish between say $0, 1, \ldots, B-1$ and $\geq B$, where B is the number of boundary layers (to be chosen). See Figure 8.2 for an illustration; shading has been used to group states that will be considered as one state for the purpose of estimating the optimal transition probabilities.



1 boundary layer 2 boundary layers 3 boundary layers

Figure 8.2: Using boundary layers to reduce the state space of a two-node queueing system.

The hardest problem is of course choosing the number of boundary layers. The only way to do this seems to be experimentation: try a low number of boundary layers and see whether the system converges to a reliable simulation (e.g., error decreasing properly with increasing number of replications); if not, increase the number of boundary layers. Looking at a graph of the transition probabilities versus the coordinates can also help: if enough boundary layers have been used, it is to be expected that the transition probability at queue content $B-1$ is not much different from the same probability at B (which actually covers $\geq B$). For more details, see the examples in Section 8.3.

8.2.3 Smoothing using spline fitting[3]

Plots of the obtained transition probabilities as a function of the buffer contents, such as Figures 8.8 and 8.10, typically show a rather smooth, monotonous change, with noise superposed on it (due to the fact that these are simulation results). Obviously, such noise in the transition probabilities used for the next iteration degrades the simulation accuracy (variance), so it is desirable to remove it.

In principle, the local-average technique should be able to achieve that. For every state, it replaces the direct estimate by an average over a set of surrounding states. In practice it was found that this is not effective enough: either too many states are grouped which degrades the simulation because the tilting then depends too weakly on the state, or the noise is not reduced sufficiently.

The spline-fitting is an attempt to reduce the noise without loosing too much of the state-dependence. The idea is that it should be possible to approximate the transition rates by a smooth function of the coordinates (the contents of the queues). Since in general the actual form of the dependence of the transition probabilities on the state is not known, functions from some rather general family need to be used, whose parameters can be adjusted to make them fit to the (noisy) simulation results.

Principle of the spline fitting

A spline is a function that is described piecewise by polynomials. This means that the domain on which the spline is to be defined is divided into segments, and on each segment a separate polynomial is defined. These polynomials are typically chosen such that the resulting spline has some form of smoothness at the segment boundaries, e.g., continuity of the first derivative. Thus, splines provide a flexible way to approximate smooth functions.

For the state-dependent importance sampling problem, the functions to be approximated by splines are the new (optimal) transition probabilities. Their domain is the state space of the Markov chain (e.g., $\mathbb{N}^n$ for a network containing n queues). Figure 8.3 shows a possible division of the state space of a two-node queueing system; every dashed rectangle indicates a piece of the domain that will be covered by one polynomial function. Note that, as before, no states are combined that have different sets of enabled transitions, i.e., states where different sets of queues are empty. Note further that most of the states are drawn in gray; only the corner states of the regions are drawn in black. This is to stress the type of approximation that will be used: in those corner points, the value

[3] Note that the technique described here is completely different from what is known as "spline smoothing" in the literature; see the remarks at the end of this section.

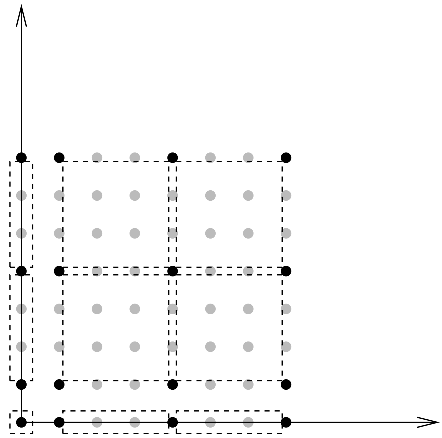


Figure 8.3: Typical division of state space for spline approximation.

of the function itself will be estimated, as well as its derivatives with respect to the coordinates (number of customers in the queues). Then the function on the region will be chosen as a polynomial which has precisely those values at the corner points. This scheme can trivially be generalized to more than the two dimensions shown here.

Base polynomials

First, consider a one-dimensional region in the above scheme. Without loss of generality, choose this to be the interval $[0, 1]$. We are given the value of the function itself, and of its first derivative, at both end points (0 and 1); and we are looking for a polynomial approximation of this function to be based on (only) this information. A polynomial that can fit this description will in general be of at least degree three; otherwise, it does not have enough degrees of freedom to match all four requirements (i.e., values of the function itself and its first derivatives at both ends of the interval).

Consider the following four degree-three polynomials (plotted in Figure 8.4), and their properties at the end points of the interval $[0, 1]$:

$f(t)$	$f(0)$	$f(1)$	$f'(0)$	$f'(1)$
$g_0(t) = 2t^3 - 3t^2 + 1$	1	0	0	0
$g_1(t) = -2t^3 + 3t^2$	0	1	0	0
$h_0(t) = t \cdot (1-t)^2$	0	0	1	0
$h_1(t) = t^2 \cdot (1-t)$	0	0	0	1

Each of these polynomials has a unity contribution to one of the (end-point) quantities which specify the function for which an approximation is sought. Con-

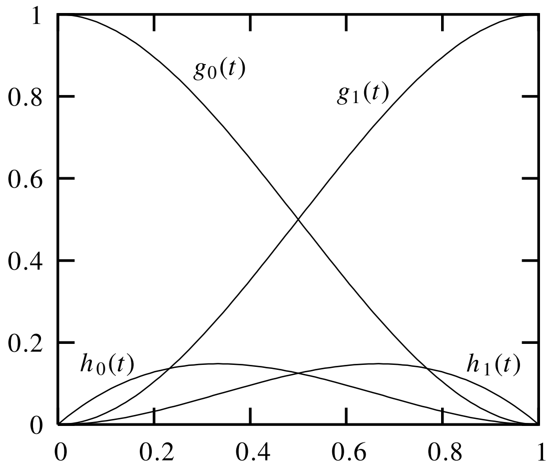


Figure 8.4: The four one-dimensional base polynomials.

sequently, if the degree-three polynomial approximation for that function is written as a weighted sum of the above *base polynomials*, the weight coefficients are precisely the given quantities. In other words: assume the function to be approximated is $z(t)$, and the values of $z(0)$, $z(1)$, $z'(0)$ and $z'(1)$ are given, then the only fitting third-order polynomial is

$$z(0)\ g_0(t) + z(1)\ g_1(t) + z'(0)\, h_0(t) + z'(1)\, h_1(t).$$

For the two-dimensional case, suitable base-polynomials can be obtained by calculating the *tensor-products* of the above one-dimensional base-polynomials. The tensor product $f(\cdot,\cdot)$ of two functions $f_1(\cdot)$ and $f_2(\cdot)$ is defined as

$$f(x,y) = f_1(x) f_2(y).$$

Substituting the one-dimensional base polynomials g_0, g_1, h_0 and h_1 into this, a set of two-dimensional base polynomials is obtained, which have properties at the corner points of $[0,1] \times [0,1]$ similar to those of the one-dimensional base-polynomials. For example, at the corner point $(0,0)$ we have:

$f(x,y)$	$f(0,0)$	$\partial_x f(0,0)$	$\partial_y f(0,0)$
$g_0(x)\cdot\ g_0(y)$	1	0	0
$h_0(x)\cdot\ g_0(y)$	0	1	0
$g_0(x)\cdot h_0(y)$	0	0	1
$h_0(x)\cdot h_0(y)$	0	0	0

Graphs of these four polynomials are shown in Figure 8.5. These four polynomials and their first derivatives are 0 at the other corner points, $(1,0)$, $(0,1)$

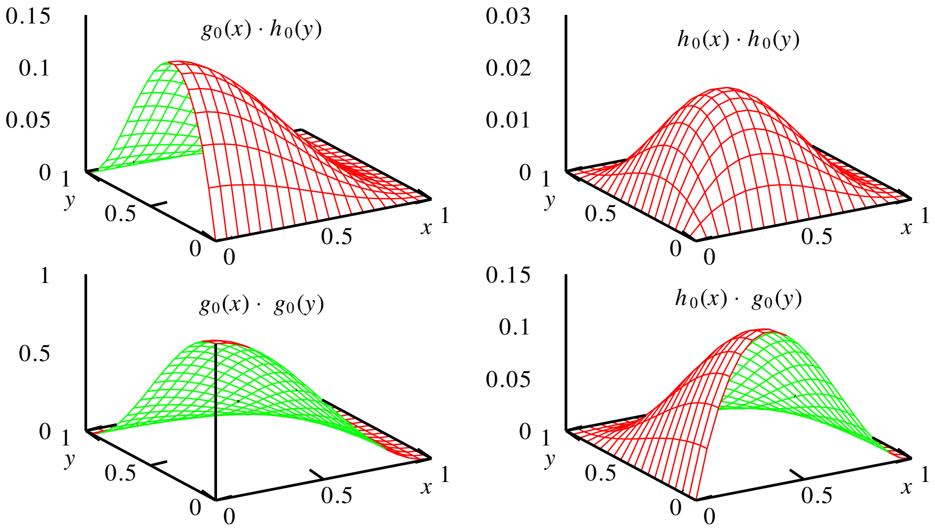


Figure 8.5: Four of the 2-dimensional base-polynomials.

and (1, 1). Similarly, all of the other twelve polynomials that can be constructed from g_0, g_1, h_0 and h_1, and their derivatives, are 0 at (0, 0). Just as in the one-dimensional case, these polynomials can be linearly combined to construct a polynomial approximation for any function whose value and first derivatives at the corners of a rectangle are given. In fact, the tensor product terms of one h polynomial with another h polynomial (such as the last one in the above table) are not even needed for this, since they have a zero contribution to the value and the first derivatives at all corner points; we set their coefficients to zero.

These tensor products can trivially be generalized to more dimensions, without losing the desired properties at the corner points.

Weighted least-squares estimation

With the base polynomials described above, we can approximate any function for which we know the value and the derivatives at a set of corner points, e.g., the black points in Figure 8.3. The only remaining problem is estimating those values and derivatives.

Estimating the value of a function and its first derivatives at a point basically means fitting a flat surface to the function, which can easily be done by the well-known least-squares algorithm. Of course, the standard least-squares algorithm would try to fit a flat surface to the entire set of data points, which is not what we need (if this were a good approximation, no splines would be needed).

Therefore, a modified version of the least-squares algorithm is used, which takes a weighting factor into account for every data point. The weighting factor should be chosen larger for points that are near the point of interest (i.e., one of the corner points of the splines), and small for points far away; typically, it would be zero for all points beyond a certain distance. This method is not new; e.g., it was also used in [CDG88].

After some experimentation, the following simple weight function was found to work well:

$$w(\boldsymbol{x},\boldsymbol{y}) = \max\left(0\ ,\ 1 - \sum_i \left(\frac{x_i - y_i}{S}\right)^2\right)$$

where S is the stepsize of the grid of the splines (3 in the example of Figure 8.3), and $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors containing the coordinates of the corner point of interest and the point whose weight is to be determined.

Choosing the corner points

In Figure 8.3, one possible division of a two-dimensional state space into regions for the spline approximation has been shown; in this example, the corner points are at coordinate values of 0, 1, 4 and 7. Choosing the corner points is a trade-off: if more corner points would be used, the splines would consist of more segments of base polynomials, so they would have more freedom to fit the real function, thus decreasing the error. However, since the weighted least squares estimation discussed above would be based on fewer points, the parameter estimates of the splines would have a larger variance, thus increasing the error.

In practice, optimizing the set of corner points is not very important: as we will see in Section 8.3, the spline method is mainly useful for speeding up the initial convergence of the iterative procedure, and is typically switched off later on.

Other smoothing methods

The spline-fitting technique described above is a rather ad-hoc smoothing technique: it is relatively simple, both conceptually and implementation-wise, and (as we will see in the experiments section) it is very effective in practice. From a theoretical point of view, however, this technique is suboptimal for the following reasons:

- The (raw) data at points that happen to be near the center of a segment on which a polynomial of the spline is defined, contribute less to the coefficients of these polynomials than the (raw) data at points near the corners. This is a consequence of the weighted least-squares approximation used to determine those coefficients. A better technique would let all data points contribute equally, or according to their accuracy if such information is available.

- The spline-fitting procedure is a *parametric* model: we assume that the real function can be well approximated by a cubic polynomial spline, and then optimize the parameters of that spline. However, since in principle we do not know anything about the real function, a *non-parameteric* smoothing technique would be more appropriate: a technique which does not make a priori assumptions about the form of the real function.

Several techniques without these problems are available in the literature on regression. One example is the "local regression" technique described in [Cle79], [CDG88] and [CD88]: this technique does a weighted least-squares approximation for *every* separate point. Some other examples are described in [Eub88], including a technique called "spline smoothing"; this is completely different from the spline-fitting described in this section, and does not have the disadvantages mentioned above. Apart from some preliminary experiments with local regression, none of these have been tried yet. An important reason for this is the fact that the spline-fitting technique works well; as we will see at the end of this chapter, the real present limitations on the usability of the adaptive state-dependent importance sampling method are not related to the imperfections of the smoothing techniques. Possibly, improved smoothing techniques will be required once these other problems will have been solved.

Disadvantage of smoothing

Whatever smoothing method is used, it puts a restriction on the form of the function. On one hand, that's precisely what is needed: restrict the function to a set of functions which are (in some sense) not "noisy". On the other hand, we do not know the form of the exact function; chances are it does not fit the smoothing restrictions. For example, it is unlikely that the optimal transition probabilies are polynomial functions of the buffer contents, but the spline method described in this section does model them as such. So approximating the real transition probabilities by a smooth function introduces some "fitting error" (but no bias for the rare event probability estimate, as noted before in the context of grouping states, see Section 8.2). Thus, a trade-off has to be made: applying smoothing reduces the error caused by the "noise" in the raw simulation results, but introduces its own (fitting) error.

This trade-off can be seen clearly in the experiments section, in Figure 8.9 for example. At relatively low numbers of replications, the raw simulation results are relatively noisy so applying spline smoothing helps: significantly fewer iterations are needed. On the other hand, when the number of replications is increased the raw simulation results become more accurate; in this case, the fitting errors from spline smoothing actually reduce the accuracy, leading to a higher variance for the resulting rare-event probability estimate.

8.2.4 Overview

To summarize, the adaptive importance sampling procedure looks like this:

1. Choose the number of boundary layers to be used; this should preferably be done in advance, since it significantly reduces memory usage.
   Also, other parameters need to be fixed in advance, such as the number of replications per iteration, maximum acceptable relative error in the local average method, etc.

2. Initialize the iteration counter $j := 1$, and the initial transition probabilities vector $\boldsymbol{\lambda}_1$ (see below).

3. Simulate N replications using transition probabilities λ_j. While doing this, keep track of

$$\sum_Z I(Z)L(Z,\boldsymbol{\lambda}_j)\sum_{i:z_i=l} 1_{(z_{i+1}=m)} \qquad \text{and} \qquad \sum_Z I(Z)L^2(Z,\boldsymbol{\lambda}_j)\sum_{i:z_i=l} 1_{(z_{i+1}=m)}$$

   for all pairs of states l and m; or rather, distinct groups of states resulting from the boundary layer technique. Also, the number of times each state / distinct group of states is visited must be recorded.

4. Apply the local average algorithm (as detailed in Section 8.2.1) to these data, yielding estimates for the simulation parameters $\boldsymbol{\lambda}_{j+1}$.

5. Optionally, apply the spline-based smoothing to $\boldsymbol{\lambda}_{j+1}$.

6. Increment the iteration counter: $j := j + 1$.

7. Repeat steps 3–6 until convergence has been achieved.

One obvious choice for the initial transition probabilities is the “original” (untilted) transition rates, like we did in the previous chapter. However, with this choice the rare event of interest will typically not be reached, so in step 3 the rare event needs to be modified as discussed in Section 7.1.3 (e.g., by lowering the overflow level). Doing so may cause a problem, because it temporarily removes some states from the state space: no transition probabilities are estimated for these states, so it is unclear what transition probabilities should be used for these states after they have been added again for the next iteration. Therefore, another approach is mostly used: initially, perform one or more iterations of the state-independent procedure described in Chapter 7, and use the obtained rates as the initial tilting $\boldsymbol{\lambda}_1$ for the above procedure. Alternatively, a heuristic and/or large-deviations based method such as [FLA91] can be used to choose the initial set of transition probabilities.

8.3 Experimental results

In this section, we investigate experimentally the performance of the adaptive state-dependent tilting method described above. We start with a trivial example, namely the $M/M/1$ queue, and then consider progressively more difficult examples. All of these examples concern the overflow probability of either the total population or a single queue in a Markovian queueing network; the latter restriction is necessary because the method at present only works for DTMC models.

As usual, the accuracy of simulation estimates is given as the relative error: the estimated standard deviation divided by the probability estimate itself.

8.3.1 The $M/M/1$ queue

The first example concerns the overflow probability in an $M/M/1$ queue. It is well-known that importance sampling with a state-independent tilting works well for estimating this probability: it provides an estimate with bounded relative error (i.e., for a given number of replications and given arrival and service rates, the relative error does not depend on the overflow level). See [PW89] and [Sad91].

Still, it is of interest to try the state-dependent method on this simple system, for several reasons:

- To check whether state-dependence gives gain compared to the state-independent method.
- The DTMC model of the $M/M/1$ queue is basically a birth-death process, and thus has a one-dimensional state space. Consequently, all states are visited on every sample path from the empty state to the overflow state. Thus, no techniques like boundary layers or local averaging are needed, because every state is visited often enough.
- Because of the one-dimensional state-space and the availability of an analytical solution to equation (8.1) (see Appendix 8.A), the state-dependent transition probabilities as obtained from the iterative simulation procedure can be easily plotted and compared with the exact values.

For the experiments, we chose the following parameters: $\lambda = 0.4$, $\mu = 0.6$, and overflow level $= 25$. Theoretically, the overflow probability with these parameters is $1.98018 \cdot 10^{-5}$. To get things started, first one iteration (with 10^3 replications) of the state-independent DTMC procedure discussed in the previous chapter was performed; this set the simulation parameters to $\lambda_1 = 0.659817$ and $\mu_1 = 0.340183$, obviously making the queue unstable. From then on, the state-dependent method was used; the resulting estimates and relative errors are shown in Table 8.1. The fifth iteration (using the state-dependent transition probabilities calculated in the fourth iteration) was performed twice, once with 10^3 and once with 10^5 replications. Continuing with the transition probabilities from the latter simulation, the sixth and the seventh iteration were performed only with 10^5 replications.

iteration	replications	estimate	rel.error
1	10^3	$2.23730 \cdot 10^{-5}$	0.140
2	10^3	$1.95122 \cdot 10^{-5}$	0.0230
3	10^3	$1.99364 \cdot 10^{-5}$	0.00602
4	10^3	$1.96880 \cdot 10^{-5}$	0.00475
5	10^3	$1.97421 \cdot 10^{-5}$	0.00474
5	10^5	$1.98058 \cdot 10^{-5}$	0.000473
6	10^5	$1.98013 \cdot 10^{-5}$	0.0000583
7	10^5	$1.98016 \cdot 10^{-5}$	0.0000401

Note: continuing the iterations with 10^3 replications gave a relative error varying between about 0.003 and 0.006; continuing with 10^5 replications similarly showed a relative error varying between about 0.00003 and 0.00006.

Table 8.1: Simulation results for the $M/M/1$ queue.

Already after few iterations, a good estimate with a low relative error has been obtained. For comparison, note that a simulation using optimal state-independent tilting with 10^3 replications gives a relative error of about 0.04, which is 10 times worse than the result with state-dependent tilting. Increasing the number of replications by a factor of 100 should of course decrease the relative error by a factor of $\sqrt{100} = 10$; indeed, the difference in the relative error between the simulations using 10^3 and 10^5 replications in iteration 5 is a factor 10. However, iterating further with 10^5 replications decreases the relative error by another factor of about 10: in total, the relative error has decreased by a factor of about 100. This is a consequence of the fact that not only the estimate of the rare-event probability itself benefits from the increased number of replications, but also the estimates of the optimal transition probabilities used in the simulation improve; this will be analysed further in Section 8.4.2.

Figure 8.6 shows the state-dependent arrival probability as a function of the state (number of customers in the queue). Four sets of data are shown: the

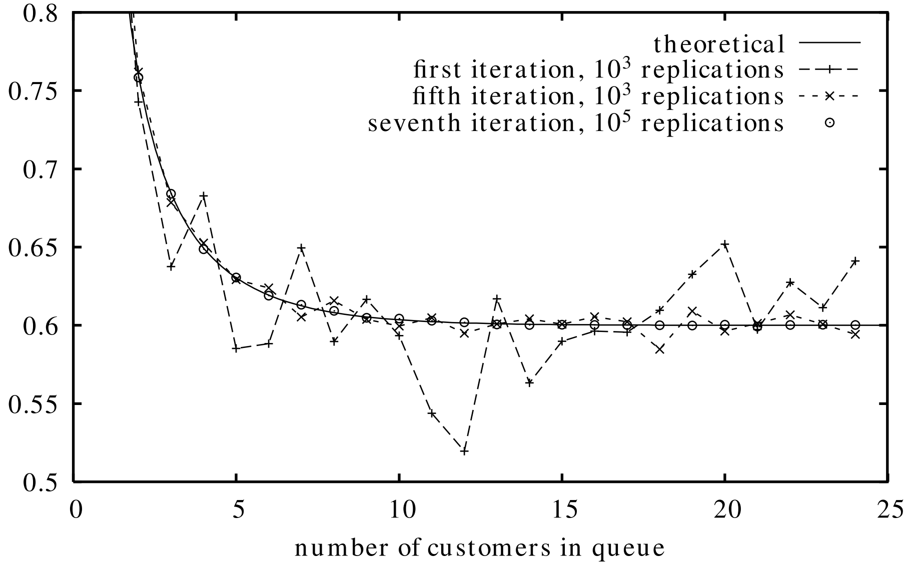


Figure 8.6: State-dependent arrival probability in the $M/M/1$ experiment.

ideal theoretical values (solid line), the practical results after the first iteration (dashed line; note the noisiness), after the fifth iteration with 10^3 replications (dotted line) and after the seventh iteration (with 10^5 replications; circles). The latter clearly agree well with the theoretical values.

8.3.2 Two queues in tandem

The second example considers two queues in tandem, with exponentially distributed interarrival and service times. The arrival rate is 0.04, both service rates are 0.48, and the rare event is the total network population reaching a high level before it reaches zero, starting also from zero. In short, this is the problem for which the adaptive state-independent tilting method turned out to work very badly in Section 7.4.4. This problem can easily be handled by the state-dependent method. We will look at a rather high overflow level, namely 100; this is twice as high as the overflow level of 50 for which the state-independent method already produced incorrect results. We start the state-dependent iterative simulation procedure with the rates resulting from one iteration of the state-independent DTMC algorithm: $\boldsymbol{\lambda}_1 = (\lambda, \mu_1, \mu_2)_1 = (0.522, 0.412, 0.066)$. Obviously, with these rates the system is unstable, so any overflow level will be reached.

Results are shown in Table 8.2. For comparison, the overflow probability is $1.3270 \cdot 10^{-105}$ according to the numerical method from Chapter 2.

The upper part of Table 8.2 was obtained using only the boundary-layers technique to reduce the state space; the other two techniques (local average and spline smoothing) were not used. Using 4 boundary layers proved sufficient, effectively reducing the state space to only $5 \times 5 = 25$ states. With 10^4 replications per iteration, the system is seen to quickly converge to a correct estimate with a low relative error. Apparently with 10^4 replications, each of the states is visited sufficiently often to give a reliable estimate of the optimal transition probabilities. The last two lines of the upper part of the table show what happens when the number of replications is increased to 10^5: the relative error decreases by approximately $\sqrt{10}$, as should be expected. On the other hand, no convergence was observed when repeating the entire experiment with 10^3 replications per iteration; apparently, with so few replications the estimates of the transition probabilities are not accurate enough.

For the lower part of the table, the local average method from Section 8.2.1 was applied (on the basis of the same 4 boundary layers), with the number of replications per iteration reduced to 10^3 (note that at 10^3 replications, no convergence was observed without the local average technique). This data is also presented graphically in the top part of Figure 8.7.

It is clear from the table that with this reduced number of replications, the convergence now takes many more iterations: convergence is achieved by the 11th as opposed to the 4th iteration. But because of the much smaller amount of work per iteration, the total simulation effort spent on achieving this convergence is lower: $11 \cdot 10^3$ instead of $4 \cdot 10^4$ replications.

The data labeled "avg. d" are the average values of d, as defined in Section 8.2.1; this is a measure for how many states were grouped on average. Apparently, in the beginning the estimates of the transition probabilities are more noisy than toward the end, so in the beginning more states need to be grouped in order to have acceptably accurate estimates of the transition probabilities; toward the end, the raw estimates become more accurate, so less grouping is needed, and the dependence of the transition probabilities on the state can be more fine-grained.

It looks like the convergence process can be divided into two phases: (a) increasing estimate, and (b) decreasing relative error. In this example, the increasing-estimate phase roughly comprises iterations 1 through 8; during this phase, most of the estimates are underestimates with a tendency to increase, and the estimated relative error varies between about 20 and 40 % (although the estimate is off by much more than this). The decreasing-relative-error phase is roughly iterations 9 through 11: the relative error decreases quickly, and the estimate approaches its correct value. After this, nothing changes significantly anymore: the system has converged. Note that during the first two phases the average d tends to decrease roughly linearly (with some noise, of course).

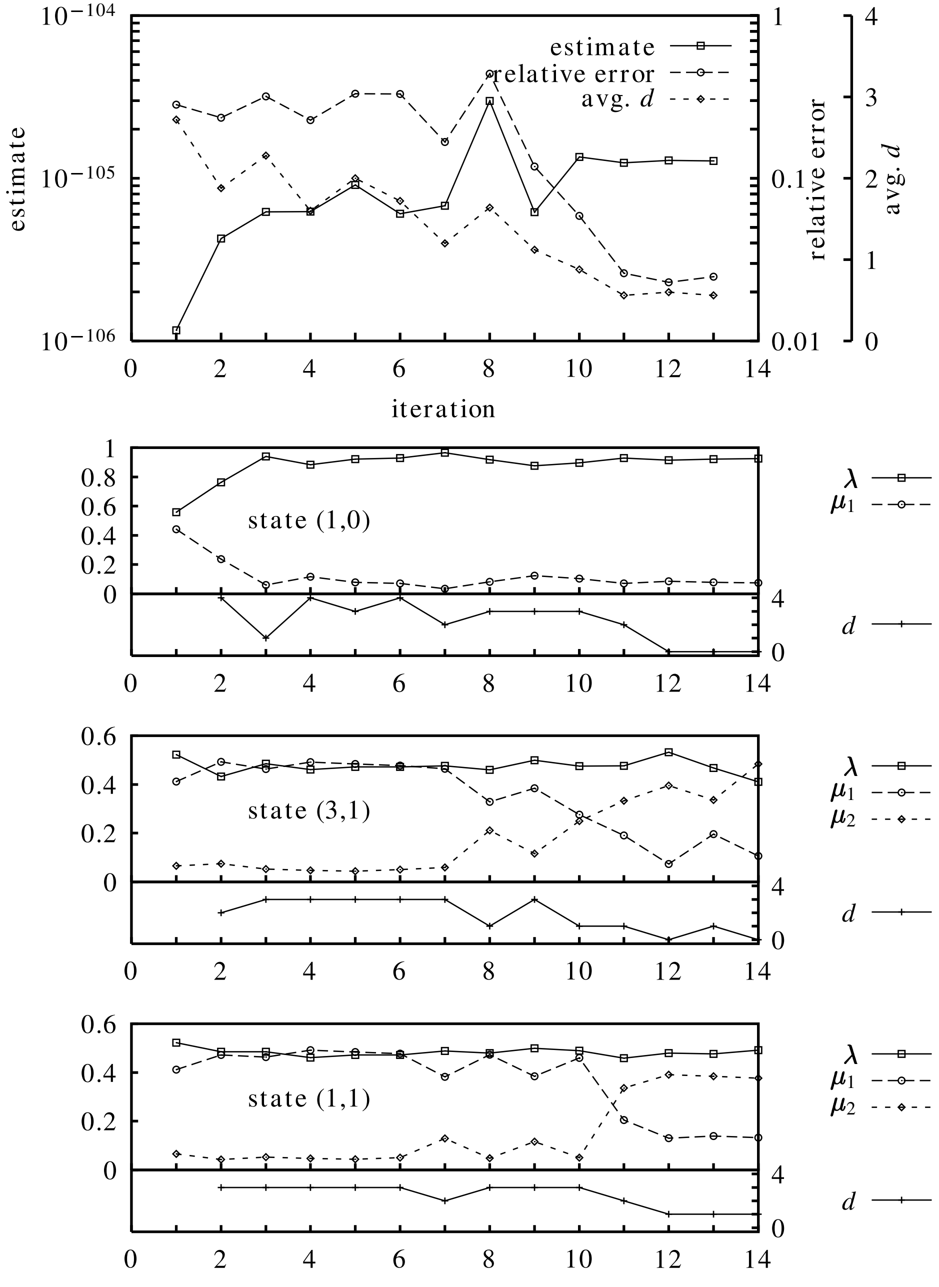


Figure 8.7: Experimental results for two queues in tandem. with 10^3 replications per iteration, using local averaging.

Without local average

iteration	replications	estimate	rel.error
1	10^4	$1.5908 \cdot 10^{-106}$	0.24655
2	10^4	$6.8313 \cdot 10^{-106}$	0.14455
3	10^4	$1.3743 \cdot 10^{-105}$	0.02378
4	10^4	$1.3281 \cdot 10^{-105}$	0.00723
5	10^4	$1.3199 \cdot 10^{-105}$	0.00709
6	10^4	$1.3162 \cdot 10^{-105}$	0.00714
6	10^5	$1.3218 \cdot 10^{-105}$	0.00226
7	10^5	$1.3281 \cdot 10^{-105}$	0.00224

With local average

iteration	replications	estimate	rel.error	avg. d
1	10^3	$0.1163 \cdot 10^{-105}$	0.28336	2.72
2	10^3	$0.4256 \cdot 10^{-105}$	0.23534	1.88
3	10^3	$0.6218 \cdot 10^{-105}$	0.3183	2.28
4	10^3	$0.6228 \cdot 10^{-105}$	0.22737	1.6
5	10^3	$0.9087 \cdot 10^{-105}$	0.33040	2.0
6	10^3	$0.6055 \cdot 10^{-105}$	0.32944	1.72
7	10^3	$0.6791 \cdot 10^{-105}$	0.16709	1.2
8	10^3	$2.9875 \cdot 10^{-105}$	0.43953	1.64
9	10^3	$0.6200 \cdot 10^{-105}$	0.11808	1.12
10	10^3	$1.3549 \cdot 10^{-105}$	0.05876	0.88
11	10^3	$1.2440 \cdot 10^{-105}$	0.02604	0.56
12	10^3	$1.2856 \cdot 10^{-105}$	0.02291	0.6
13	10^3	$1.2769 \cdot 10^{-105}$	0.02488	0.56
13	10^4	$1.3019 \cdot 10^{-105}$	0.00792	0.24
14	10^4	$1.3246 \cdot 10^{-105}$	0.00743	0.24

Table 8.2: Experimental results for two queues in tandem.

The lower part of Figure 8.7 shows how the three transition probabilities vary in the course of the iterations[4], for some selected states (namely (1,0), (3,1) and (1,1), as indicated in the graphs); it is of course not practical to show such graphs for all 25 states. Clearly, in some states the transition probabilities converge earlier to their final values than they do in some other states. Also, the value of d of these states is plotted: this tells us whether the transition probabilities are a relatively local estimate, or an estimate based on grouping many states. There is

[4]Note that these are the values of the transition probabilities *used* at the *present* iteration, i.e., *resulting* from the *previous* iteration. Thus, the transition probabilities plotted at iteration number 1 are simply the initial state-independent probabilities.

clearly a correlation between the lowering of d (thus a more local estimate) and the convergence of the transition probabilities to their final values, as was to be expected.

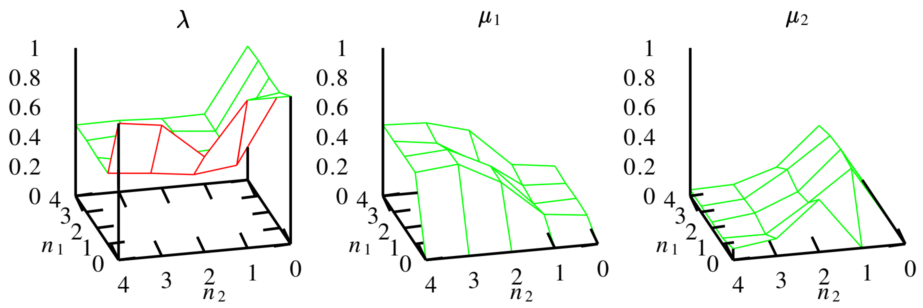


Figure 8.8: State-dependent transition probabilities for two queues in tandem.

Finally, Figure 8.8 shows the dependence of the three transition probabilities on the levels of the two buffers (n_1 and n_2), as obtained after the 6th iteration without the local average method. Note that due to the application of the boundary layer method, level 4 actually represents all levels ≥ 4. The fact that at $n_1 = 3$ or $n_2 = 3$ the probabilities are not much different from those at $n_1 \geq 4$ or $n_2 \geq 4$, respectively, is an indication that 4 boundary layers are indeed enough. In this particular case, the graphs suggest that queue 1 could have done with fewer boundary layers (since for $n_1 \geq 2$ the probabilities hardly change with n_1), but queue 2 could not.

8.3.3 Four queues in tandem

In this example, we consider a network consisting of four queues in tandem. As in the previous example, the parameters are chosen in the region where the standard state-independent tilting (exchanging the arrival rate with the bottleneck service rate) does not work well according to [GK95]: the arrival rate is 0.09, and the service rates of the first through fourth queue are 0.23, 0.227, 0.227 and 0.226, respectively. The rare event of interest is again the total network population reaching a high level, starting from 0, and before returning to 0 again.

Results for overflow level 50

The results for an overflow level of 50 are presented in Figure 8.9. For the curves in the left part of the graph, the local-average method and the boundary-layer method (with 10 boundary layers) were used; for the curves in the right part, additionally the spline smoothing was used[5]. Up to the 23rd iteration (without splines) and the 9th iteration (with splines), each iteration contained 10^4 replications.

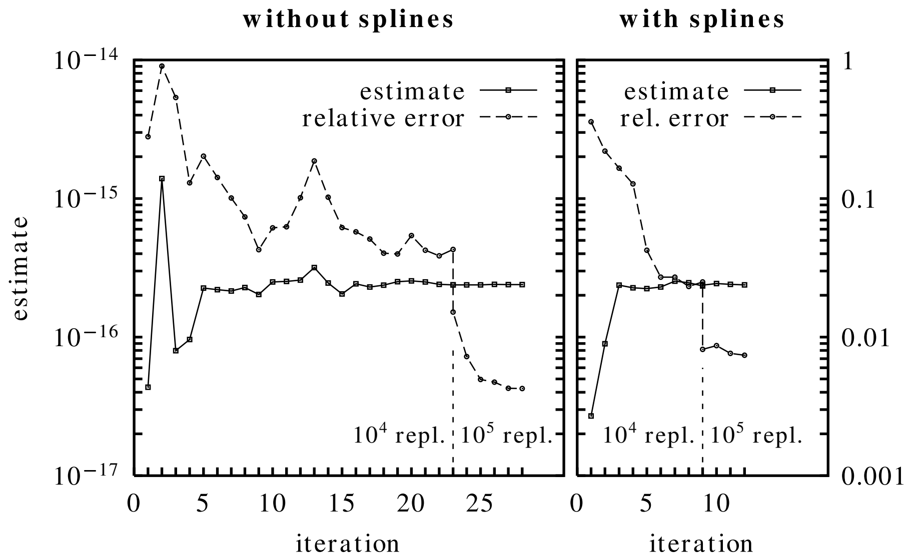
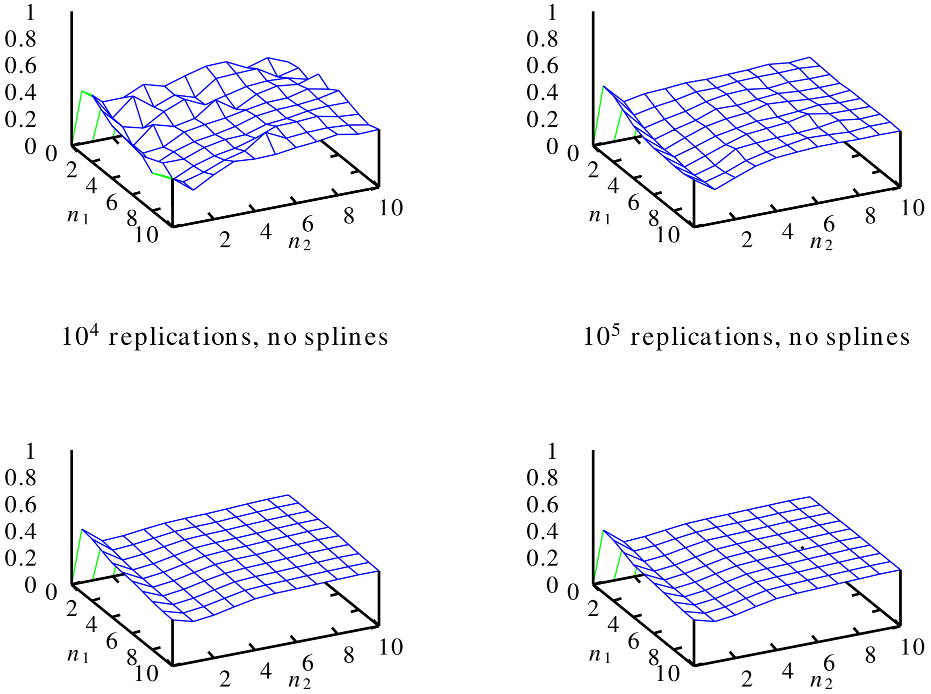


Figure 8.9: Results for four queues in tandem, overflow level = 50.

Obviously, the spline method is quite beneficial to the convergence in this case. The without-splines part of the graph shows a rather slow and irregular convergence, with a major excursion around the 13th iteration, whereas with the splines method the convergence is quick and monotonous, and the resulting relative error is smaller by almost a factor of 2.

Furthermore, note what happens when the number of replications is increased. The 23rd (without splines) and the 9th (with splines) simulation were performed twice: once with 10^4 replications, and once with 10^5 replications; the iterations were continued with 10^5 replications. Without splines, the same effects as observed in Section 8.3.1 are seen here: at first, the relative error decreases by about $\sqrt{10}$, but upon iterating further, it decreases by a factor of 10 in the end; again, the cause of this is the improved estimation of the transition probabilities. With splines this effect does not happen, and the final error with 10^5 replications is larger with splines than without; apparently, the spline form does not fit the true functions well enough to allow a further decrease of the relative error. See also the discussion of the "fitting error" on page 150.

[5]The spline basepoints (see Section 8.2.3) were set at levels 0, 1, 5 and 10.



10^4 replications, no splines

10^5 replications, no splines

10^4 replications, with splines

10^5 replications, with splines

Figure 8.10: State-dependent transition probabilities (service completion at first queue) for four queues in tandem ($n_3 = n_4 = 0$).

In the two-queue example, plots of the state-dependent transition probabilities as functions of the state were presented (Figure 8.8). Doing the same for the present example is not feasible, because its state space is four-dimensional instead of just two-dimensional. At best, a two-dimensional "slice" of the state-space can be plotted, and this is done in Figure 8.10. All these plots show the transition probability corresponding to service completion at the first queue as a function of the content of the first and second queues, while the third and the fourth queues are empty. Clearly, the splines perform a very effective smoothing: most of the noise disappears. On the other hand, the splines used here are apparently not able to completely follow the true functions: the "dip" at $n_2 = 1$ is much deeper without splines (only sufficiently visible in the 10^5-replications plot) than with splines. This agrees with the experimental observation that at 10^5 replications, the final estimate is more accurate when the state-dependence

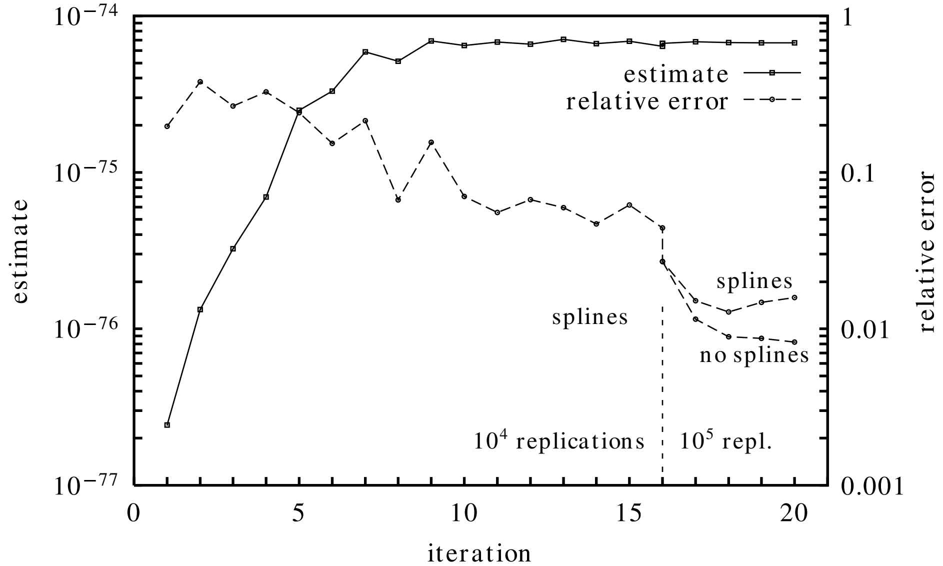


Figure 8.11: Results for four queues in tandem, overflow level = 200.

of the transition probabilities is not restricted by applying splines.

Results for overflow level 200

For the case of an overflow level of 200, Figure 8.11 shows the simulation results. For this problem, all three techniques (local average, 10 boundary layers, and splines) were used initially (up to iteration 16), with 10^4 replications per iteration. After convergence had been achieved with this, the iterations were continued with 10^5 replications each, and both with and without splines, resulting in two branches in the graph. Some points are noteworthy:

First, it takes about seven iterations before the estimate is near the correct value; during those iterations, it increases monotonously. Thus, the increasing-estimate phase (as it was called in the previous example) is quite pronounced here. Actually, it is somewhat surprising that while the rare-event probability of interest is estimated completely wrong, the procedure still converges. Consecutive iterations are only linked by the transition probabilities, so it seems that these are still estimated more or less correctly.

Second, at 10^5 replications, the splines still work quite well, but switching them off makes the relative error smaller. This is again the trade-off between "fitting errors" and simulation noise, discussed on page 150.

iteration	replications	estimate	rel.error	avg. d
1	10^5	$6.3321 \cdot 10^{-15}$	0.5375	3.25
2	10^5	$4.9917 \cdot 10^{-15}$	0.0254	0.53
3	10^5	$5.1039 \cdot 10^{-15}$	0.0091	0.65
4	10^5	$5.0925 \cdot 10^{-15}$	0.0093	0.71
5	10^5	$5.1688 \cdot 10^{-15}$	0.0092	0.56
5	10^6	$5.2128 \cdot 10^{-15}$	0.0030	0.25
6	10^6	$5.1991 \cdot 10^{-15}$	0.0029	0.25
7	10^6	$5.1838 \cdot 10^{-15}$	0.0029	0.25
8	10^6	$5.1811 \cdot 10^{-15}$	0.0029	0.22

Table 8.3: Results for the three bounded queues.

8.3.4 Three queues in tandem with bounded buffers

The following system was considered as an example for RESTART simulation in [Gar00]. It comprises three queues in tandem. Each of the queues is bounded, and each server has an exponentially distributed service time. The interarrival time at the first queue is also exponentially distributed. The system starts in a state with one customer in the first and the second queue, and none in the third queue. The rare event probability of interest is the probability that the third queue reaches a given high level before becoming empty for the first time.

The following parameter values are used: arrival rate = 1, service rate of first and second queue = 2, service rate of third queue = 4, size of first buffer (including the customer being served) = 40, second buffer = 20, and overflow level of the third queue = 20. Note that the third queue, whose overflow probability is to be determined, is *not* the bottleneck queue.

Since we are not looking for overflow of the total network population or of the bottleneck queue, the standard heuristic of exchanging the arrival rate with the bottleneck service rate is not applicable. Furthermore, the adaptive state-independent tilting method from Chapter 7 turns out to be not effective either, so we resort to state-dependent tilting. Four boundary-layers and the local-average techniques were used. For every iteration, 10^5 replications were simulated. To get things started, two iterations with the state-independent method were performed first; Table 8.3 shows the results for the subsequent state-dependent iterations. For comparison, the exact probability is $5.1863 \cdot 10^{-15}$, according to the method from Chapter 2. The agreement between the simulation results and this numerical result clearly is good.

8.3.5 A five-node Jackson network with feedback and routing

As a final example, consider the overflow probability of the total population in the network of five queues shown in Figure 8.12. Customers arrive according to a Poisson process with rate λ, and the service times are exponentially distributed with rates μ_1 through μ_5, as indicated in the drawing. Furthermore, at the

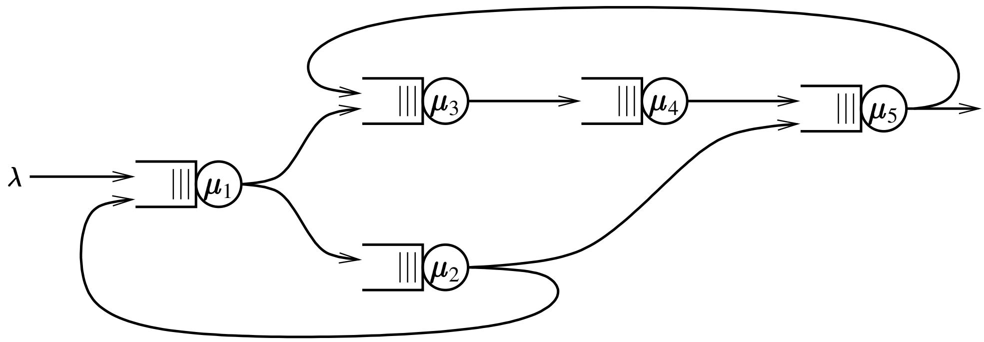


Figure 8.12: A five-node Jackson network.

outputs of queues 1, 2 and 5, random routing takes place: customers are equally likely to choose one of two routes. We use the following rates:

$$\lambda = 3 \quad \mu_1 = 40 \quad \mu_2 = 20 \quad \mu_3 = 50 \quad \mu_4 = 50 \quad \mu_5 = 60$$

Note that this choice makes the load of every queue 0.1. Generally, state-independent tilting seems to work badly for systems with several equally loaded queues, and indeed, the adaptive state-independent method does not work well with this problem.

However, applying the state-dependent method is not trivial either. Because of the large number of queues, it is desirable to use a low number of boundary layers, in order to keep the state space manageable. An experiment using 4 boundary layers and the local-average method to estimate the overflow probability of level 50 failed, both with 10^4 and with 10^5 replications per iteration: after few iterations, several of the transition probabilities tended to zero, and the average value of d tended to 4 (which is the maximum when using 4 boundary layers).

Fortunately, another approach turned out to work surprisingly well: first use 10^4 replications per iteration to estimate the overflow probability of level 20, and then use the obtained state-dependent transition probabilities as a starting point for estimating the overflow probability of level 50. For the latter estimation, 10^4 replications per iterations turned out to be insufficient, so 10^5 were used

iteration	overflow level	replications	boundary layers	estimate	relative error
1	20	10^4	4	$7.263 \cdot 10^{-16}$	0.1323
2	20	10^4	4	$9.100 \cdot 10^{-16}$	0.1011
3	20	10^4	4	$8.326 \cdot 10^{-16}$	0.0703
4	20	10^4	4	$7.352 \cdot 10^{-16}$	0.0280
5	20	10^4	4	$7.752 \cdot 10^{-16}$	0.0247
6	50	10^5	4	$2.720 \cdot 10^{-44}$	0.0473
7	50	10^5	4	$2.556 \cdot 10^{-44}$	0.0186
8	50	10^5	4	$2.600 \cdot 10^{-44}$	0.0103
9	50	10^5	4	$2.619 \cdot 10^{-44}$	0.0100
10	50	10^5	4	$2.612 \cdot 10^{-44}$	0.0079
11	50	10^5	4	$2.643 \cdot 10^{-44}$	0.0123
12	50	10^5	4	$2.591 \cdot 10^{-44}$	0.0078
13	50	10^5	4	$2.586 \cdot 10^{-44}$	0.0079
13	50	10^6	4	$2.605 \cdot 10^{-44}$	0.0025
14	50	10^6	4	$2.609 \cdot 10^{-44}$	0.0015
15	50	10^6	4	$2.610 \cdot 10^{-44}$	0.0014
15	100	10^5	7	$3.979 \cdot 10^{-93}$	0.0174
16	100	10^5	7	$4.174 \cdot 10^{-93}$	0.0288
17	100	10^5	7	$3.982 \cdot 10^{-93}$	0.0274
18	100	10^5	7	$3.761 \cdot 10^{-93}$	0.0194
19	100	10^5	7	$3.935 \cdot 10^{-93}$	0.0150
20	100	10^5	7	$3.940 \cdot 10^{-93}$	0.0208
21	100	10^5	7	$3.954 \cdot 10^{-93}$	0.0158
22	100	10^5	7	$4.045 \cdot 10^{-93}$	0.0257
23	100	10^5	7	$3.846 \cdot 10^{-93}$	0.0175
23	100	10^6	7	$3.921 \cdot 10^{-93}$	0.0061
24	100	10^6	7	$3.951 \cdot 10^{-93}$	0.0036
25	100	10^6	7	$3.955 \cdot 10^{-93}$	0.0024
26	100	10^6	7	$3.950 \cdot 10^{-93}$	0.0019
27	100	10^6	7	$3.964 \cdot 10^{-93}$	0.0017
28	100	10^6	7	$3.968 \cdot 10^{-93}$	0.0016
29	100	10^6	7	$3.949 \cdot 10^{-93}$	0.0016
30	100	10^6	7	$3.959 \cdot 10^{-93}$	0.0015
31	100	10^6	7	$3.969 \cdot 10^{-93}$	0.0017

Table 8.4: Results for the five-node Jackson network.

(and later 10^6, for extra accuracy and verification). After convergence had been achieved for overflow level 50, overflow level 100 was tried. For this, 10^5 and even 10^6 replications per iteration were not sufficient: the relative error kept increasing; also using the splines approximation did not help. What did help, however, was increasing the number of boundary layers to 7, resulting in a small relative error again, even with 10^5 replications. Table 8.4 shows the results. Note: for getting things started, two iterations of the state-independent method were used (with overflow level 20 and 10^4 replications per iteration); these are (as usual) not shown in the table.

The above example shows that temporarily making the target event less rare (in this case by lowering the overflow level) can be beneficial to help the iterative method to "find" the optimal tilting. Temporarily lowering the target event's rarity has been discussed before, in Sections 8.2.4 and 7.1.3, and originally in [Rub97]. However, the purpose in those cases was different: lowering the rarity served to ensure that the event would be observed often enough. In the present case, the event would be observed often enough even if we would not lower the overflow level (because the first iteration already makes the system unstable, so any high level would be reached often).

In Section 8.2.4, it was noted that temporarily lowering an overflow level could produce implementation problems, since it is unclear what transition probabilities should be assigned to the states that are added when the overflow level is raised again. In the present example, this problem did not occur, mostly due to the application of the boundary-layer technique: this technique groups most of the states removed by the lower overflow level with states that were not removed and for which transition probabilities could still be estimated.

8.3.6 Asymptotic efficiency

Table 8.5 shows the simulation results (and, as far as available, numerical results according to Chapter 2) for some of the systems discussed above for several different values of the overflow levels. The relative error typically grows no more than linearly with the overflow level, while the probability estimate decreases exponentially. Thus, the method is asymptotically efficient, at least in all of these examples.

8.4 Mathematical foundations

In this section, the adaptive state-dependent method will be studied mathematically. A complete description will not be given. But under some simplifying assumptions, interesting insight can still be gained.

model	level	exact	estimate	rel.error
two queues in tandem (10^4 repl.)	12	$1.469 \cdot 10^{-11}$	$1.462 \cdot 10^{-11}$	0.0041
	25	$2.872 \cdot 10^{-25}$	$2.879 \cdot 10^{-25}$	0.0057
	50	$6.033 \cdot 10^{-52}$	$5.988 \cdot 10^{-52}$	0.0068
	100	$1.327 \cdot 10^{-105}$	$1.325 \cdot 10^{-105}$	0.0074
	150	$2.188 \cdot 10^{-159}$	$2.177 \cdot 10^{-159}$	0.0074
four queues in tandem (10^5 repl.)	25	$3.528 \cdot 10^{-7}$	$3.504 \cdot 10^{-7}$	0.0026
	50	–	$2.396 \cdot 10^{-16}$	0.0042
	100	–	$1.422 \cdot 10^{-35}$	0.0044
	200	–	$6.722 \cdot 10^{-75}$	0.0082
three bounded queues (10^5 repl.)	10	$1.183 \cdot 10^{-7}$	$1.182 \cdot 10^{-7}$	0.0045
	20	$5.186 \cdot 10^{-15}$	$5.120 \cdot 10^{-15}$	0.0096
	40	$1.762 \cdot 10^{-29}$	$1.761 \cdot 10^{-29}$	0.0189
	80	–	$1.448 \cdot 10^{-58}$	0.0520
	160	–	$1.133 \cdot 10^{-116}$	0.1605

Table 8.5: Test of asymptotic efficiency.

First, we will show that in principle, the method tends to converge to precisely that set of transition probabilities which would give a zero variance estimation of the quantity of interest. These considerations will lead to an alternative view of how the method works.

Second, the influence of the statistical error in the transition probabilities (due to the fact that they are simulation results themselves) will be investigated. This leads to an explanation for the experimental observation that the variance of the rare-event probability estimator can decrease proportional to the *square* of the number of replications used.

8.4.1 Zero variance

Start by rewriting the denominator of (8.1) as follows:

$$\begin{aligned}\mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1 &= \sum_{i=1}^{\infty} \mathbb{E}_0 I(Z) 1_{(z_i=l)} = \sum_{i=1}^{\infty} \mathbb{P}(I(Z)=1 \ \wedge \ z_i = l) \\ &= \sum_{i=1}^{\infty} \mathbb{P}(I(Z)=1 \mid z_i = l)\mathbb{P}(z_i = l) = \sum_{i=1}^{\infty} \pi_l \mathbb{P}(z_i = l),\end{aligned}$$

where π_l is defined as the probability of reaching the rare event before an absorbing state is reached, starting from state l; note that π_{z_1} is the rare-event probability of interest. Rewrite the numerator similarly:

$$
\begin{aligned}
&\mathbb{E}_0 I(Z) \sum_{i:z_i=l} 1_{(z_{i+1}=m)} \\
&= \sum_{i=1}^{\infty} \mathbb{E}_0 I(Z) 1_{(z_i=l)} 1_{(z_{i+1}=m)} \\
&= \sum_{i=1}^{\infty} \mathbb{P}(I(Z)=1 \wedge z_i = l \wedge z_{i+1} = m) \\
&= \sum_{i=1}^{\infty} \mathbb{P}(I(Z)=1 \mid z_i = l \wedge z_{i+1} = m) \cdot \mathbb{P}(z_{i+1} = m \mid z_i = l) \cdot \mathbb{P}(z_i = l) \\
&= \sum_{i=1}^{\infty} \pi_m p_{lm} \mathbb{P}(z_i = l),
\end{aligned}
$$

where p_{lm} is the untilted transition probability from state l to state m. By substituting the above into (8.1), we find the following expression for the optimal transition probabilities:

$$
q_{lm} = \frac{\lambda^{\dagger}_{lm}}{\sum_k \lambda^{\dagger}_{lk}} = \frac{\sum_{i=1}^{\infty} \pi_m p_{lm} \mathbb{P}(z_i = l)}{\sum_{i=1}^{\infty} \pi_l \mathbb{P}(z_i = l)} = \frac{\pi_m p_{lm} \sum_{i=1}^{\infty} \mathbb{P}(z_i = l)}{\pi_l \sum_{i=1}^{\infty} \mathbb{P}(z_i = l)} = \frac{\pi_m p_{lm}}{\pi_l}. \tag{8.5}
$$

One can easily recognise the right-hand side as the conditional probability of going from state l to state m, given that the rare event will be reached.

Now consider a random path Z leading to the rare event, containing n_Z steps. The probability of this path in the tilted simulation is

$$
\prod_{i=1}^{n_Z} q_{z_i z_{i+1}} = \frac{p_{z_1 z_2} \pi_{z_2}}{\pi_{z_1}} \cdot \frac{p_{z_2 z_3} \pi_{z_3}}{\pi_{z_2}} \cdots \frac{p_{z_{n_Z} z_{n_Z+1}} \pi_{z_{n_Z+1}}}{\pi_{z_{n_Z}}} = \frac{\pi_{z_{n_Z+1}}}{\pi_{z_1}} \prod_{i=1}^{n_Z} p_{z_i z_{i+1}}.
$$

The probability of the same path in the untilted system is just $\prod_{i=1}^{n_Z} p_{z_i z_{i+1}}$, so the likelihood ratio is $\pi_{z_1}/\pi_{z_{n_Z}}$. Since Z was defined as a path leading to the rare event, its last state z_{n_Z+1} must be the rare event itself; therefore $\pi_{z_{n_Z+1}} = 1$. Consequently, the likelihood ratio of the path is just π_{z_1}, which is (by definition) the rare-event probability, and thus a constant independent of the path. Since *all* sample paths in the tilted system reach the rare event (see[6] the last paragraph of Section 8.1.4), and thus have this same likelihood ratio, the variance of the estimator is zero[7].

[6] Or observe that the expectation (in the tilted system) of $I(Z)L(Z)$ is the probability of interest π_{z_1}. We have just shown that, given $I(Z) = 1$, L itself equals π_{z_1}, so the expectation of $I(Z)L(Z)$ can be equal to π_{z_1} only if $I(Z) = 1$ for any sample path in the tilted system (remember that $I(Z)$ is an indicator function and thus is either 0 or 1). Thus, every sample path in the tilted system reaches the rare event.

[7] Zero-variance estimators in Markov chains have been discussed before in the literature; e.g., Section VI.2.4 in [Erm75].

In practice, zero variance is not reached of course. There are two reasons for this:

- The new transition probabilities are estimated from simulation results, which obviously have a statistical error; the effect of this is the subject of the next section. Note that this error can be made arbitrarily small by increasing the number of replications used in the simulation.

- The techniques for dealing with the large state space discussed in Section 8.2 introduce errors. E.g., the boundary-layer method makes all transition probabilities far away from the boundaries equal, and the spline approximation only allows transition probabilities which fit the form of the splines used; these errors don't disappear with increasing number of replications. The local average technique is different in this respect: with increasing number of replications, fewer states need to be grouped, so the error introduced also decreases.

An alternative view of the method

Up to here, the adaptive state-dependent importance sampling simulation method was basically considered as a way to automatically find the optimal parameters for reducing the variance in an importance-sampling simulation. An alternative view is possible, however.

Let us consider what would happen if indeed the optimal values of the transition probabilities were found. In that case, all sample paths leading to the rare event would have the same likelihood ratio, as shown above; that likelihood ratio would in fact be equal to the probability of interest (π_{z_1}). So simulating only *one* sample path would already provide us with a perfect estimate of the rare-event probability of interest. Thus, the problem has been converted from a problem where the quantity of interest is simulated directly, to a problem where the conditional transition probabilities are obtained through simulation, and the quantity of interest is obtained through a calculation (namely, following one sample path, which doesn't need to be random) based on those simulation results.

In practice, approximations to the optimal transition probabilities are obtained by simulation, and thus subject to statistical errors. These errors cause the likelihood ratio along sample paths to the rare event to become variable, so in order to get an accurate estimation of the rare-event probability, the likelihood ratios from many sample paths need to be averaged; this is what the simulation in the next iteration does.

8.4.2 Influence of the statistical error in the transition probability estimates

As discussed above, the non-zero variance of the rare-event probability estimator is caused by the statistical error in the transition probability estimates. In the following, a model of these statistical errors is constructed, which is subsequently used to calculate the resulting variance of the rare-event probability estimator. This serves to give insight into the relative importance of the iteration which estimates the transition probabilities, and the subsequent iteration which uses these to estimate the rare-event probability.

For clarity, let us refer to the simulation iteration in which the rare-event probability is estimated[8] as the "last" simulation (or iteration). Then this simulation uses transition probabilities which are simulation results from the "second-last" simulation. On its turn, the second-last iteration uses transition probabilities which are simulation results from the third-last simulation, and so on.

As before, define q_{lm} as the optimal transition probability from state l to m. Furthermore, define $\hat{q}_{lm}$ as the simulation estimate of this probability (obtained in the second-last iteration), which contains a statistical error. Assume that this error has a normal distribution[9] with relative variance σ^2_{lm}, as follows:

$$\hat{q}_{lm} = q_{lm} \cdot (1 + f_{lm}) \qquad \text{with} \qquad f_{lm} \sim \mathcal{N}\big(0, \sigma^2_{lm}\big).$$

Using the optimal transition probabilities q_{lm}, the likelihood ratio would be a constant (independent of the sample path), but using the simulation estimates $\hat{q}_{lm}$, this is not the case. The ratio of the ideal (L_Z) and the practical ($\hat{L}_Z$) likelihood ratio on a sample path Z is

$$\begin{aligned}\frac{L_Z}{\hat{L}_Z} &= \prod_{i=1}^{n_Z}(1 + f_{z_i z_{i+1}}) = 1 + \sum_{i=1}^{n_Z} f_{z_i z_{i+1}} + \sum_{i=1}^{n_Z}\sum_{j=1}^{n_Z} f_{z_i z_{i+1}} f_{z_j z_{j+1}} + \ldots \\ &\approx 1 + \sum_{i=1}^{n_Z} f_{z_i z_{i+1}} \sim \mathcal{N}\big(1, \sum_{i=1}^{n_Z} \sigma^2_{z_i z_{i+1}}\big),\end{aligned}$$

where it is assumed that typically $\sigma_{lm} < 1/n_Z$, so the higher order terms can indeed be neglected at the $\approx$ sign. Furthermore, the above assumes that the errors f_{lm} are independent of each other, and that the sample path does not (or rarely)

[8] Of course, one can estimate the rare-event probability (as a by-product) in every iteration, and this was in fact done in the experiments in Section 8.3. However, for the purpose of discussing the estimate's variance, we need to refer clearly to one estimate; that's why this definition is made.

[9] Although a normal distribution is usually an appropriate approximation for a simulation error, it is not completely realistic here. It does not take into account the fact that all transition probabilities from a state sum up to 1. Furthermore, it does not take into account that all transition probabilities must be between 0 and 1: the normal distribution has infinite tails. However, for small variances these effects can be neglected.

visit a state twice; otherwise, the variance of the resulting normal distribution would be different due to the dependencies.

We see that the variance of the likelihood ratio (which is the variance of the estimator of the rare-event probability, since all sample paths reach the rare event) is proportional to the variance of the individual transition probabilities. Since these individual transition probabilities are simulation results from the second-last simulation, their variance is inversely proportional to the number of replications used in that simulation. Clearly, the variance of the rare-event probability estimator is also inversely proportional to the number of replications used in the last simulation. Thus, we find that the variance of the final estimator is inversely proportional to the *product* of the number of replications used in the last and in the second-last simulation; or, if the same number of replications is used in both of them, the variance is inversely proportional to the *square* of the number of replications. This phenomenon has already been observed experimentally in Sections 8.3.1 and 8.3.3.

One might be tempted to take this reasoning a step further: the accuracy of the transition probabilities used in the last iteration not only depends on the number of replications in the second-last iteration, but also on the accuracy of the transition probabilities used in that second-last simulation; the latter depends on the number of replications of the third-last iteration, so the accuracy of the rare-event probability estimation should also depend on the number of replications in the third-last iteration. This can indeed be the case, but it will not nearly be as strong as the dependence on the number of replications in the second-last iteration. The reason for this is that the estimation of the transition probabilities is inherently *not* a zero-variance simulation: even if the third-last iteration had yielded perfect estimates of the transition probabilities to be used in the second-last simulation, the latter would not give zero-variance estimates of the transition probabilities for the last simulation.

8.5 Conclusions and outlook

In this chapter, an adaptive importance sampling method has been proposed in which the tilting is allowed to depend on the state of the system. This is a natural extension of the state-independent tilting method from the previous chapter, and it turns out to work very well, in particular in the cases in which the state-independent method fails. The method has turned out to be asymptotically efficient in all examples considered. Furthermore, all experiments done with this method show good agreement with results from other methods (numerical calculations in particular), whenever those are available.

A unique and somewhat unexpected property of the method is the rate at which the variance of the estimator decreases with increasing number of replications (spread over several iterations): in many cases, it decreases proportional to the square of the total number of replications, instead of linearly. This has both been observed experimentally, and justified analytically.

A few miscellaneous issues are worth noting. First, in this chapter the method has only been used to estimate probabilities of the form reaching a (rare) overflow state before reaching some other (absorbing) state; however, in regenerative systems, this can be used as a basis for finding steady-state probabilities. Second, the method is well suited for parallel processing, since all the replications (typically thousands) within an iteration are independent of one another. Third, the extra CPU time needed to do one iteration of the state-dependent instead of the state-indepent method, is usually small: the simulation itself is hardly different, so extra time would mainly be needed for the local average and spline calculations; but in almost all of the experiments, this turned out to take only a small fraction of the time needed for the simulation itself.

Besides all its good properties, the method still has its limitations: it only applies to DTMC models, and the state space must not be too large. These problems and possible solutions are discussed in more detail below.

8.5.1 Further improvements for large state space

The techniques discussed in Section 8.2 have been shown to be quite effective for queueing network models containing up to about five queues. However, without enhancements these techniques are bound to fail with larger networks: after application of the boundary layer method the state space still contains $(1+b)^n$ states, where b is the number of boundary layers used, and n the number of queues. Typically, 4 to 10 boundary layers are used, so for every additional queue the state space increases by a factor of 5 to 11, and the amount of memory required increases proportionally. Furthermore, the CPU time needed for the local average and spline calculations also increase with the size of the state space.

One possible way to deal with this problem is suggested by plots like Figure 8.8 and 8.10: these show that typically the state-dependent transition probabilities do not depend strongly on all coordinates (queue contents), but only on some of them. For the coordinates on which the probabilities do not (or hardly) depend, fewer (possibly zero) boundary layers could be used, thus effectively reducing the state space. Unfortunately, graphs like the ones mentioned above are only available *after* the procedure has converged, so they cannot be used to help the convergence by reducing the state space in advance. However, further investigation of these dependencies for some typical examples may give more insight to decide in advance which coordinates are important.

8.5.2 Extension to non-Markovian models

Some non-Markovian models can easily be converted into DTMC models by extending the state space, so it contains (discrete) state information about the non-exponential distribution involved. For example, this is possible if the model contains phase-type distributions (such as Erlang and hyperexponential). In such cases, the techniques discussed thus far can be applied to the DTMC version of the model.

If non-phase-type distributions, such as deterministic or uniform, are involved, it is no longer possible to reduce the model to a DTMC with a discrete state-space. To fully describe the state of such a model, one or more *continuous* variables are needed. In such models, two types of problems arise when trying to apply the adaptive state-dependent tilting methods discussed in this chapter:

- Representation of the tilting as a function of continuous variables.
- Sampling from a random variable whose (tilted) distribution changes with time due to the changing state.

These issues are discussed in more detail below. Some suggestions to deal with the problems are given, but, due to time limitations, they have not yet been implemented.

Representation of the tilting as a function of continuous variables

If inter-arrival and service times have non-exponential (and non-phase-type) distributions, or e.g. delays need to be estimated, the state of a queueing system is no longer adequately described by just the number of customers in every queue (and the phase of phase-type distributions). Instead, also the elapsed times since the last arrival and/or service completions need to be stored; these are continuous random variables, so they can take an (uncountably) infinite number of values: the state space becomes (uncountably) infinite. Possible approaches to deal with this include:

- Just ignore it: make the tilting independent of the elapsed times, so it only depends on the discrete state variables.
- Partition the continuous state space into a finite number of bins, and make the tilting depend only on the bin (which is a discrete variable).
- Model the state-dependent tilting as some function of the (continuous) state variable. This could be similar to the spline fitting from Section 8.2.3.

Clearly, these approaches are listed in order of increasing complexity and decreasing "coarseness" of the dependence of the tilting on the (continuous) variable.

Sampling from a random variable with a changing distribution

The usual way of simulating a non-DTMC system is an event-based mechanism: at some point (e.g., upon an arrival to an empty queue) a sample is taken from some distribution (e.g., the service-time distribution of that queue), from which the time for some future event (e.g., service completion at that queue) is calculated (e.g., "now" + "service time sample"), and the event is recorded in an event list. Then, other (earlier) events in the event list are processed, before the event considered above is finally processed.

Naïvely, a state-dependent tilting could be implemented simply by using the tilting belonging to the current state at the moment the sample is taken. However, this may not be optimal: between the taking of a sample, and the processing of the event scheduled based on that sample, the state of the system can change quite significantly. It would be preferable if the sample could somehow be updated when the system state changes. This could be done by *resampling* the random variable every time the system changes; a suitable procedure is described in [NNHG90]. An alternative approach is *uniformization*, as decribed in [NHS92]: in this approach, no scheduling in the usual sense is used. Instead, one keeps track of which events have been scheduled since when, and this information is used to choose the next event to happen with the correct (time-dependent) probabilities.

Appendix

8.A Exact optimal transition probabilities for $M/M/1/k$

In this appendix, the optimal transition probabilities as given by (8.1) are calculated exactly for the $M/M/1/k$ queue.

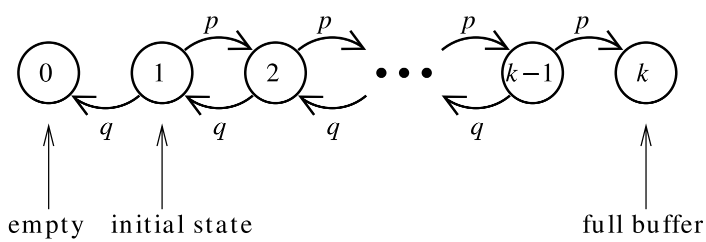


Figure 8.13: The birth-death process corresponding to an $M/M/1/k$ queue.

Model the $M/M/1/k$ queue as a birth-death process, as illustrated in Figure 8.13. The states range from 0 (empty state, regeneration state) up to k (full buffer). The probability of an arrival (i.e., jump to the next higher state) is $p = \lambda/(\lambda + \mu)$, and of a service completion (jump to the next lower state) is $q = 1 - p$.

In Section 8.4.1, this equation for the optimal probabilities has been rewritten as (8.5), which expresses the optimal probabilities in terms of the "untilted" probabilities p_{lm} and the probabilities π_l of reaching the rare event before emptying, starting from state l. The following analytical expression for π_l can be found from Section 2.2.1 or [NH96]:

$$\pi_l = \frac{\left(\frac{\mu}{\lambda}\right)^l - 1}{\left(\frac{\mu}{\lambda}\right)^k - 1}.$$

Substituting this into (8.5) we find for the optimal arrival probability in state l:

$$p_l^\dagger = p\frac{\left(\frac{\mu}{\lambda}\right)^{l+1} - 1}{\left(\frac{\mu}{\lambda}\right)^l - 1} = \frac{q^{l+1} - p^{l+1}}{q^l - p^l},$$

where the fact that $\mu/\lambda = q/p$ was used. The optimal service probability follows from

$$q_l^\dagger = 1 - p_l^\dagger = 1 - \frac{q^{l+1} - p^{l+1}}{q^l - p^l}.$$

Note that these optimal transition probabilities are independent of the overflow level k.

A plot of $p_l^\dagger$ for various values of the load (p/q) of the queue is shown in Figure 8.14. Clearly, unless the queue is heavily loaded, the optimal transition probabilities differ from the large-deviations result ($p^* = q$, $q^* = p$) significantly only in the states close to the empty state.

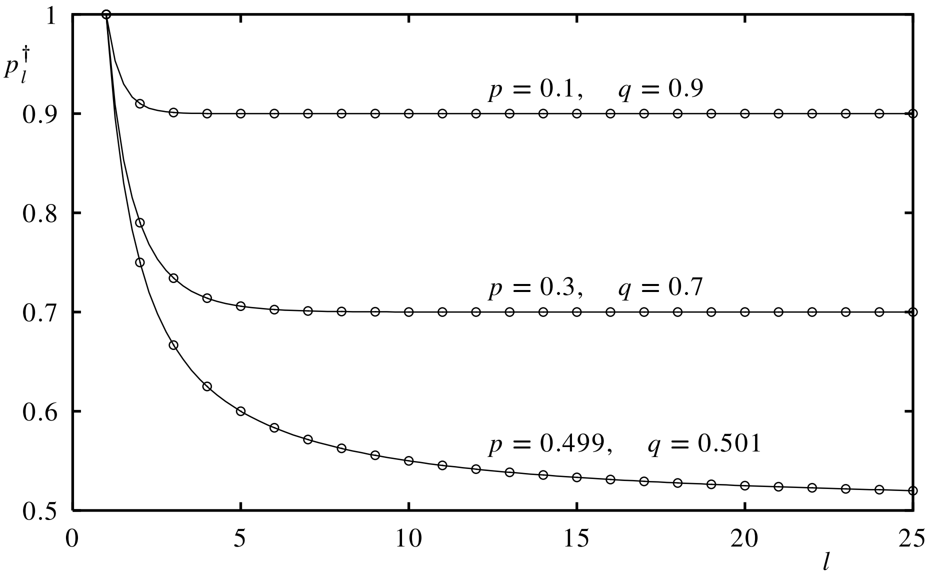


Figure 8.14: Optimal state-dependent arrival probability $p^\dagger$.

Chapter 9

Conclusions

In this chapter, the main contributions of this thesis are listed and some possible future extensions of the work are discussed.

9.1 Main contributions

Consecutive cell loss

One problem studied in this thesis is the estimation of consecutive (cell) loss probabilities and frequencies in simple queueing models. This subject has received little attention in the literature, and this was mostly limited to models of very specific ATM subsystems. The present thesis has focused on more general queueing models, resulting in the following contributions:

- Analytical (numerical) calculation for $M/G/1$ and $G/M/m$ queues.
- Analytical (numerical) calculation for one stream in a multiple-stream $M/M/1$ queue.
- Asymptotically efficient (for large number of consecutive cells lost) importance sampling simulation for $M/G/1$ queues where the service time distribution is upper-bounded. This is actually a hybrid approach: analysis is used to express the probability of interest in terms of four other probabilities, each of which is estimated by simulation. Alternatively, these can also be approximated by asymptotic analysis.

Overflows in networks of queues

The estimation of overflow probabilities in queueing networks has received considerable attention in the importance sampling simulation literature. Most of the literature has concentrated on heuristically derived changes of measure, which perform well in many, but not all, models. Adaptive methods (i.e., methods which try to iteratively approach the optimal change of measure) have only been applied to queueing problems in [DT93a], in which a different adaptive method is used than in the present work, and [RM98], where only a few simple models are considered.

In this thesis, several adaptive importance sampling simulation methods for the estimation of overflow probabilities in queueing networks are developed (on the basis of [RM98]) and their performance is compared. These methods can be classified as follows:

- State-independent tilting via variance minimization
- State-independent tilting via cross-entropy minimization
- State-independent tilting via cross-entropy minimization, for Markovian models
- State-dependent tilting via cross-entropy minimization, for Markovian models

The first three of these use a state-independent change of measure: the tilting does not depend on the state of the system (i.e., number of customers in the queues). These three methods differ in whether they explicitly minimize the simulation variance, or do this indirectly by minimizing a related quantity (the cross-entropy), which has some computational advantages and turns out to give equally good results. The third method contains some enhancements specific for Markovian models. It is found experimentally that these three methods generally work quite well, although some counterexamples are shown in which irregularities or even completely wrong estimates are observed. For these counterexamples, no asymptotically efficient state-independent change of measure seems to exist.

The last method allows the change of measure to depend on the state. This makes the change of measure much more flexible; as a consequence, better simulation performance is obtained. On the other hand, determining such a change of measure is more complicated, especially if the state space is large; several techniques to deal with this problem have been discussed. The results with this method are consistently good: asymptotically efficient estimation of overflow probabilities is possible even for models in which the methods with a state-independent change of measure do not work well. The method has only been fully developed for Markovian models; possible solutions to the additional complexities of non-Markovian models have been discussed briefly.

For verification of the simulation results, a simple way to numerically calculate overflow probabilities in Jackson networks has also been demonstrated.

Other results

In the course of the research, several sub-problems have been solved; the results could also have applications in other contexts. These results include:

- Asymptotic expressions for the past and remaining service times upon reaching a high level (full buffer) in $M/G/1$ queues; besides the applications to the estimation of consecutive cell loss probabilities demonstrated in this thesis, it has also been applied in RESTART simulations [Gar00].
- Asymptotically efficient (for large n) simulation methods for the estimation of probabilities of the form $\sum_i^n X_i < Y$; such probabilities play a role in many practical problems, including reliability models, signal detection and queueing.
- An extension of the central limit theorem to exponentially tilted random variables; this is useful in asymptotic efficiency proofs for importance sampling schemes.
- The numerical evaluation method for overflow probabilities in Jackson networks; it was developed here for the validation of simulation results, but can of course be applied in practical queueing network models of telecommunications systems, manufacturing systems, etc.; furthermore, it can be applied to discrete-time Markov chains arising in non-queueing contexts.

9.2 Future work

Cell loss patterns

Some extension of the work presented here on consecutive cell loss is of interest, e.g., extension of the analysis to other queues than $M/G/1$ and $G/M/m$, and to more complicated per-stream models than just $M/M/1$. Developing a provably asymptotically efficient simulation method for systems with unbounded service time distributions would also be of interest, but this runs into a fundamental problem: it would be necessary to change the distribution of the duration of the full buffer periods (cf. Chapter 6), but since that duration is a simulation result itself, it cannot readily be changed. The results regarding i.i.d. sums derived in this thesis may still be useful for developing an efficient simulation method for consecutive cell loss in queues with bounded service times but non-Markovian interarrival times.

However, for practical applications it would probably be of more interest to focus on more general cell loss patterns, such as the probability of losing m out of n consecutive cells. Such events can have a similar influence on the Quality of Service of a telecommunications system as the loss of m consecutive cells, and they are more likely.

Overflows in networks of queues

Two obvious important directions for future research are extension of the state-dependent tilting method to non-Markovian systems, and development of better techniques for handling very large state spaces. Some ideas for this have already been mentioned at the end of Chapter 8.

For practical applications, the simulation of models with many (independent) non-Poisson sources is important. The importance sampling simulation as used in Chapter 7 is not good at handling such models, if the sources are just implemented and tilted independently. This problem is not specific to the adaptive methods, but it does limit their applicability in realistic models; therefore, finding a good way to handle such models is important.

Finally, it is desirable to develop a better mathematical understanding of the working of the adaptive methods considered in this thesis. For example, it is not understood well why the methods converge, particularly in the case of state-dependent tilting. Also, it is not known whether the minimum cross-entropy tilting is always close (and perhaps asymptotically identical) to the minimum-variance tilting.

Bibliography

[AQDT95] Wael A. Al-Qaq, Michael Devetsikiotis, and J. Keith Townsend. Stochastic gradient optimization of importance sampling for the efficient simulation of digital communication systems. *IEEE Transactions on Communications*, 43:2975–2985, 1995.

[Asm81] Søren Asmussen. Equilibrium properties of the $M/G/1$ queue. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 58:267–281, 1981.

[BHJ60] E. Brockmeyer, H. L. Halstrøm, and A. Jensen. The life and works of A.K. Erlang. Acta Polytechnica Scandinavica AP 287, 1960.

[Bon91] André B. Bondi. On the bunching of cell losses in ATM-based networks. In *GLOBECOM 91*, pages 0444–0447, 1991.

[CD88] William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83:596–610, September 1988.

[CDG88] William S. Cleveland, Susan J. Devlin, and Eric Grosse. Regression by local fitting — methods, properties and computational algorithms. *Journal of Econometrics*, 37:87–114, 1988.

[CFM83] Marie Cottrell, Jean-Claude Fort, and Gérard Malgouyres. Large deviations and rare events in the study of stochastic algorithms. *IEEE Transactions on Automatic Control*, 28(9):907–920, 1983.

[Cle79] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, December 1979.

[Coh82] J. W. Cohen. *The single server queue.* North-Holland, Amsterdam, 2nd edition, 1982.

[dBN98] Pieter-Tjerk de Boer and Victor F. Nicola. Hybrid importance sampling estimation of consecutive cell loss probability. *AEÜ International Journal of Electronics and Communications*, 52:133–140, 1998.

[dBNR00a] Pieter-Tjerk de Boer, Victor F. Nicola, and Reuven Y. Rubinstein. Adaptive importance sampling simulation of queueing networks. Accepted for the Winter Simulation Conference '00, Orlando, FL, 2000.

[dBNR00b] Pieter-Tjerk de Boer, Victor F. Nicola, and Reuven Y. Rubinstein. Dynamic importance sampling simulation of queueing networks: An adaptive approach based on cross-entropy. Accepted for the Third Workshop on Rare Event Simulation, Pisa, 2000.

[dBNS99] Pieter-Tjerk de Boer, Victor F. Nicola, and Rajan Srinivasan. On the estimation of a rare event probability involving IID sums. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 133–138, 1999.

[dBNS00] Pieter-Tjerk de Boer, Victor F. Nicola, and Rajan Srinivasan. On the estimation of a rare event probability involving IID sums. Submitted to ACM Transactions on Modeling and Computer Simulation, 2000.

[dBNvO98] Pieter-Tjerk de Boer, Victor F. Nicola, and Jan-Kees C. W. van Ommeren. The remaining service time upon reaching a high level in $M/G/1$ queues. CTIT Technical Report 98-12, University of Twente, 1998. Submitted to QUESTA.

[DT93a] Michael Devetsikiotis and J. Keith Townsend. An algorithmic approach to the optimization of importance sampling parameters in digital communication system simulation. *IEEE Transactions on Communications*, 41:1464–1473, 1993.

[DT93b] Michael Devetsikiotis and J. Keith Townsend. Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks. *IEEE/ACM Transactions on Networking*, 1:293–305, 1993.

[Erm75] S. M. Ermakow. *Die Monte-Carlo-Methode und verwandte Fragen*. R. Oldenbourg Verlag, München Wien, 1975.

[Eub88] Randall L. Eubank. *Spline Smoothing and Nonparametric Regression*. Marcel Dekker, New York and Basel, 1988.

[Fak82] D. Fakinos. The expected remaining service time in a single server queue. *Operations Research*, 30:1014–1018, 1982.

[Fel66] W. Feller. *An introduction to probability theory and its applications*. Wiley, New York, 1966.

[FLA91] Michael R. Frater, Tava M. Lennon, and Brian D.O. Anderson. Optimally efficient estimation of the statistics of rare events in queueing networks. *IEEE Transactions on Automatic Control*, 36:1395–1405, 1991.

[Gar00] Marnix J. J. Garvels. *The splitting method in rare event simulation*. PhD thesis, University of Twente, 2000. In preparation.

[GF98] Carmelita Goerg and Oliver Fuß. Comparison and optimization of RESTART run time strategies. *AEÜ International Journal of Electronics and Communications*, 52:197–204, 1998.

[GHSZ98] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. A large deviations perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control*, 43(12):1666–1679, 1998.

[GK95] Paul Glasserman and Shing-Gang Kou. Analysis of an importance sampling estimator for tandem queues. *ACM Transactions on Modeling and Computer Simulation*, 5(1):22–42, January 1995.

[GK99] Marnix J. J. Garvels and Dirk P. Kroese. On the entrance distribution in RESTART simulation. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 65–88, 1999.

[GW97] Paul Glasserman and Yashan Wang. Counterexamples in importance sampling for large deviations probabilities. *The Annals of Applied Probability*, 7(3):731–746, 1997.

[Hee98a] Poul E. Heegaard. *Efficient simulation of network performance by importance sampling*. PhD thesis, Norwegian University of Science and Technology (NTNU Trondheim), 1998.

[Hee98b] Poul E. Heegaard. A scheme for adaptive biasing in importance sampling. *AEÜ International Journal of Electronics and Communications*, 52:172–182, 1998.

[Hei95] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5:43–85, 1995.

[K^+95] D.P. Kroese et al. Consecutive cell loss in queueing systems. Written communication, University of Twente, 1995.

[KK92] J. N. Kapur and H. K. Kesavan. *Entropy Optimization Principles with Applications*. Academic Press, 1992.

[KL91] W. David Kelton and Averill M. Law. *Simulation modeling and analysis*. McGraw-Hill, New York, 1991.

[Kle64] L. Kleinrock. *Communication nets: stochastic message flow and delay*. McGraw-Hill, New York, 1964.

[Kle75a] Leonard Kleinrock. *Queueing systems*, volume 1: Theory. Wiley-Interscience, 1975.

[Kle75b] Leonard Kleinrock. *Queueing systems*, volume 2: Computer applications. Wiley-Interscience, 1975.

[KN99] Dirk P. Kroese and Victor F. Nicola. Efficient simulation of a tandem Jackson network. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 197–211, 1999.

[KS97] Latha Kant and William H. Sanders. Analysis of the distribution of consecutive cell losses in an ATM switch using stochastic activity networks. *International Journal of Computer Systems Science and Engineering*, 12(2):117–129, March 1997.

[LA96] Chaewoo W. Lee and Mark S. Andersland. Consecutive cell loss controls for leaky-bucket admission systems. In *Proceedings of Globecom '96*, pages 1732–1738, 1996.

[Lie99] Dmitrii Lieber. *The cross-entropy method for estimating probabilities of rare events*. PhD thesis, William Davidson Faculty of Industrial Engineering and Management, Technion, Israel, 1999.

[LR00] Dmitrii Lieber and Reuven Y. Rubinstein. Rare-event estimation via cross-entropy and importance sampling. In preparation, 2000.

[Man96] Michel Mandjes. *Rare Event Analysis of Communication Networks*. PhD thesis, Vrije Universiteit Amsterdam, 1996.

[MR00] Michel Mandjes and Ad Ridder. A large deviations analysis of the transient of a queue with many markov fluid inputs: approximations and fast simulation. *ACM Transactions on Modeling and Computer Simulation*, 2000. To appear.

[Nak92] M. K. Nakayama. Efficient methods for generating some exponentially tilted random variates. In *Proceedings for the 1992 Winter Simulation Conference*, pages 603–608, 1992.

[Neu81] M. F. Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Johns Hopkins University Press, Baltimore, 1981.

[NH96] V. F. Nicola and G. A. Hagesteijn. Efficient simulation of consecutive cell loss in ATM networks. In D. Kouvatsos, editor, *Performance Modelling and Evaluation of ATM Networks, Vol. II*. Chapman and Hall, London, 1996.

[NHS92] Victor F. Nicola, Philip Heidelberger, and Perwez Shahabuddin. Uniformization and exponential transformation: techniques for fast simulation of highly dependable non-markovian systems. In *The Twenty-Second Annual International Symposium on Fault-Tolerant Computing*, pages 130–139, 1992.

[NNHG90] Victor F. Nicola, Marvin K. Nakayama, Philip Heidelberger, and Ambuj Goyal. Fast simulation of dependability models with general failure, repair and maintenance processes. In *Proceedings of the twentieth international symposium on fault-tolerant computing*, pages 491–498, 1990.

[NSH00] Victor F. Nicola, Perwez Shahabuddin, and Philip Heidelberger. Techniques for fast simulation of highly dependable systems. In preparation, 2000.

[Oct] `http://www.che.wisc.edu/octave/`.

[Onv94] Raif O. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*. Artech House, Boston, London, 1994.

[PW89] S. Parekh and J. Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control*, 34:54–66, 1989.

[RB00] David Remondo Bueno. *Performance Evaluation of Communication Systems via Importance Sampling*. PhD thesis, University of Twente, 2000.

[RM98] Reuven Y. Rubinstein and Benjamin Melamed. *Modern Simulation and Modeling*. Wiley, New York, 1998.

[RMV96] James Roberts, Ugo Mocci, and Jorma Virtamo, editors. *Broadband Network Teletraffic; Performance Evaluation and Design of Broadband Multiservice Networks; Final Report of Action COST 242*. Springer, Berlin [etc.], 1996.

[Rub97] Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112, 1997.

[Rub99] Reuven Y. Rubinstein. Rare event simulation via cross-entropy and importance sampling. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 1–17, 1999.

[Sad91] J. S. Sadowsky. Large deviations theory and efficient simulation of excessive backlogs in a $GI/GI/m$ queue. *IEEE Transactions on Automatic Control*, 36:1383–1394, 1991.

[Sri98a] Rajan Srinivasan. Estimation and approximation of densities of i.i.d. sums via importance sampling. *Signal Processing*, 71:235–246, 1998.

[Sri98b] Rajan Srinivasan. Some results in importance sampling and an application to detection. *Signal Processing*, 65:73–88, 1998.

[SW95] A. Shwartz and A. Weiss. *Large deviations for performance analysis*. Chapman and Hall, New York, 1995.

[Tit52] E. C. Titchmarsh. *Theory of Functions*. Oxford University Press, London, 1952.

[VAVA91] Manuel Villén-Altamirano and José Villén-Altamirano. RESTART: A method for accelerating rare event simulations. In J. W. Cohen and C. D. Pack, editors, *Proceedings of the 13th International Teletraffic Congress, Queueing Performance and Control in ATM*, pages 71–76. North Holland, 1991.

[VAVA99] Manuel Villén-Altamirano and José Villén-Altamirano. About the efficiency of RESTART. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 99–128, 1999.

[vHK68] D. van Hemert and J. Kuin. *Automatische telefonie*. Vereniging van Technisch Hoger Personeel der PTT, 1968.

[Wei] Eric W. Weisstein. Eric Weisstein's World of Mathematics. `http://mathworld.wolfram.com/`.

[Wei95] Alan Weiss. An introduction to large deviations for communication networks. *IEEE Journal on Selected Areas in Communications*, 13(6):938–952, 1995.

Summary

In modern packet-switched telecommunication systems, information (such as e-mail, sound, pictures) is transported in the form of small packets (or cells) of data through a network of links and routers. The Quality of Service provided by such a network can suffer from phenomena such as loss of packets (due to buffer overflow) and excessive delays. These aspects of the system are adequately described by queueing models, so the study of such models is of great relevance for designing systems such that they provide the required QoS. This thesis contributes methods for the efficient estimation of several loss probabilities in various queueing models of communications systems. The focus is on rare-event simulation using importance sampling, but some analytical, asymptotic and numerical results are also provided.

One part of this thesis is concerned with issues related to the estimation of the probability of consecutive (cell) loss: the loss of several consecutive arrivals to a queue. Analytical calculation of this is demonstrated for several simple queues ($M/G/1/k$ and $G/M/m/k$), and an importance sampling simulation procedure is provided for $M/G/1/k$ queues. Furthermore, an $M/M/1/k$ queue with multiple sources is considered, in which the probability of consecutive (cell) loss incurred by one of these sources is calculated analytically.

The other part of this thesis is concerned with the estimation of overflow probabilities in queueing networks. For estimating these probabilities, importance sampling simulation methods are considered, in which several adaptive techniques (mostly based on cross-entropy) are applied to approximate the optimal change of measure. Two classes of change of measure are used: those which do not depend explicitly on the state of the model (e.g., a “static” change of the arrival and service rates), and those which do (e.g., changing the arrival and service rates separately for each state). The methods using a state-independent change of measure turn out to be quite effective and to result in an asymptotically efficient simulation in most cases; however, some counterexamples are also observed. With a state-dependent change of measure, an asymptotically efficient simulation is obtained in every example tried, including those for which no good state-independent change of measure is known. The state-dependent method

has only been applied to Markovian networks, but possible ways to extend it to non-Markovian networks are briefly discussed. Furthermore, a simple numerical method is proposed for the calculation of overflow probabilities in simple Jackson networks, which is used to verify the correctness of the results from the above simulation methods.

In the course of the work on the above two main problems, some interesting subproblems and related issues were investigated. The obtained results are also useful in other contexts, and include the following: (1) asymptotic expressions for the past and remaining service time distributions upon reaching a high (overflow) level in $M/G/1$ queues; (2) asymptotically efficient importance sampling simulation schemes for the estimation of probabilities of events of the form $\sum_{i=1}^{n} X_i < Y$, where X_i are positive i.i.d. random variables, and Y is also a positive random variable (useful in e.g. reliability models); (3) an extension of the central limit theorem to exponentially tilted random variables (useful for asymptotic efficiency proofs).

Samenvatting

In moderne, pakket-geschakelde telecommunicatiesystemen wordt informatie (zoals e-mail, geluid, beelden) getransporteerd in de vorm van kleine datapakketjes (of "cellen") door een netwerk van verbindingen en routers. De Quality-of-Service (kwaliteit van de telecommunicatiedienst) die door zo'n netwerk wordt geleverd kan verslechteren door verschijnselen zoals het verloren raken van pakketjes (ten gevolge van overstroming van buffers) en onwenselijk grote vertragingen. Deze aspecten van het systeem worden goed beschreven door wachtrijmodellen; daarom is de studie van dergelijke modellen zeer relevant voor het zodanig ontwerpen van de systemen dat de telecommunicatiedienst met de gewenste kwaliteit wordt gerealiseerd. Dit proefschrift beschrijft een aantal methoden voor het efficiënt schatten van verlies-kansen in diverse wachtrijmodellen van communicatiesystemen. De nadruk ligt op de simulatie van zeldzame gebeurtenissen met behulp van importance-sampling, maar enkele analytische, asymptotische en numerieke resultaten komen ook aan de orde.

Een deel van dit proefschrift is gewijd aan zaken die te maken hebben met het schatten van de kans op "consecutive loss": het verliezen van meerdere opeenvolgend bij het wachtsysteem aankomende cellen. Een analytische berekening van deze kans wordt gegeven voor diverse elementaire wachrijen ($M/G/1/k$ and $G/M/m/k$), evenals een simulatieprocedure gebaseerd op importance sampling voor $M/G/1/k$. Bovendien wordt een $M/M/1/k$-wachtsysteem met meerdere bronnen beschouwd; in dit systeem wordt de kans op verlies van opeenvolgende cellen van één van deze bronnen analytisch berekend.

Het andere deel van dit proefschrift gaat over de schatting van overstromingskansen in netwerken van wachtsystemen. Hiertoe worden op importance-sampling gebaseerde simulatiemethoden beschouwd, waarbij diverse adaptieve technieken (meestal gebaseerd op cross-entropy) worden toegepast om de optimale kansmaatverandering te benaderen. Twee klassen van kansmaatverandering worden beschouwd: die welke niet expliciet van de toestand van het model afhangen (bijv. een "statische" verandering van de aankomst- en bedieningssnelheid), en die welke wel expliciet van de toestand van het model afhangen (bijv. een aparte wijziging van de aankomst- en bedieningssnelheid voor elke toe-

stand). De methoden met een toestandsonafhankelijke kansmaatverandering blijken behoorlijk effectief te zijn, en resulteren in de meeste gevallen in een asymptotisch efficiënte simulatie; er zijn echter ook enkele tegenvoorbeelden gevonden. Gebruik van een toestandsafhankelijke kansmaatverandering leidt in alle geteste gevallen tot een asymptotisch efficiënte simulatie, ook in die gevallen waarin geen goede toestandsonafhankelijke kansmaatverandering bekend is. De toestandsafhankelijke methode is alleen maar toegepast op Markovse netwerken, maar enkele mogelijkheden om de methode uit te breiden tot niet-Markovse netwerken worden kort besproken. Verder beschrijft dit proefschrift een eenvoudige numerieke methode om overstromingskansen in eenvoudige Jacksonnetwerken te berekenen; deze methode is gebruikt om de resultaten van bovengenoemde simulatiemethoden te controleren.

In de loop van het werk aan de twee bovengenoemde hoofdproblemen zijn ook nog enkele andere interessante subproblemen en gerelateerde zaken onderzocht. De daarbij verkregen resultaten zijn ook nuttig in andere contexten: (1) asymptotische uitdrukkingen voor de kansverdeling van de verstreken en resterende bedieningsduur op het moment dat de inhoud van een $M/G/1$ wachtsysteem een hoog niveau bereikt; (2) asymptotisch efficiënte simulatiemethoden gebaseerd op importance-sampling voor het schatten van kansen van gebeurtenissen van de vorm $\sum_{i=1}^{n} X_i < Y$, waar X_i positieve, onafhankelijke maar identiek verdeelde stochastische variabelen zijn, en Y ook een positieve stochastische variabele is (toepasbaar in bijv. betrouwbaarsheidsmodellen); (3) een uitbreiding van de centrale limiet-stelling naar exponentieel getilte stochastische variabelen (toepasbaar in bewijzen van asymptotische efficiëntie).

Dankwoord / Acknowledgements / Tankwurd

Aan het einde van dit proefschrift wil ik graag allen bedanken die op een of andere wijze aan de totstandkoming ervan hebben bijgedragen.

Om te beginnen gaat mijn dank uit naar de leden van de vakgroep TIOS en later de leerstoel TSS. In het bijzonder wil ik mijn promotor Ignas Niemegeers, en mijn dagelijks begeleider Victor Nicola bedanken. Laatstgenoemde in het bijzonder voor de vele gedetailleerde werkbesprekingen, en voor de gaandeweg steeds grotere vrijheid die ik kreeg om het onderzoek in te richten. Verder bedank ik Marnix Garvels, met wie ik gedurende ruim vier jaar de werkkamer heb gedeeld; onze gesprekken heb ik als nuttig, interessant maar zeker ook gezellig ervaren.

Ook wil ik de leden van de vakgroep STOR van Toegepaste Wiskunde bedanken voor de samenwerking, en met name Jan-Kees van Ommeren voor zijn significante bijdrage aan hoofdstuk 4.

Next, I would like to thank all the people working in the field, with whom I have had the pleasure of collaborating, who are on the committee, and/or who have provided me with useful comments on the thesis. I thank Rajan Srinivasan, who contributed to Chapter 6. I am particularly grateful to Reuven Rubinstein; discussions and collaboration with him have formed the basis for Chapters 7 and 8.

Vervolgens wil ik graag de leden van de radio-amateurvereniging ETGD bedanken. De ETGD was gedurende deze jaren een belangrijke bron van ontspanning en gezellige bijeenkomsten. Bovendien is mijn interesse voor digitale telecommunicatienetwerken vooral gewekt door het amateur-packetradionetwerk; daarom wil ik in het bijzonder Eduard van Dijk bedanken, voor de wijze waarop hij mij destijds met packetradio heeft laten kennismaken.

Oan it ein, mar seker net it minst, wol ik myn famylje, en natuerlik foaral myn âlders, betankje foar har oanhâldende steun.

Over de auteur

Pieter-Tjerk de Boer werd in 1972 geboren in Wildervank. In 1990 behaalde hij het Gymnasium-B-diploma aan het Ichthus College in Enschede. Van 1990 tot 1996 studeerde hij Technische Natuurkunde (met elektrotechnische bijvakken) aan de Universiteit Twente; zijn afstudeeropdracht bij de vakgroep Theoretische Natuurkunde was getiteld "Soliton switching in directional couplers". Daarna verrichtte hij bij de vakgroep Tele-Informatica en Open Systemen, later de leerstoel Telematics Systems and Services, van de Faculteit der Informatica van de Universiteit Twente zijn promotieonderzoek dat resulteerde in dit proefschrift. Momenteel is hij als post-doc bij deze zelfde groep werkzaam.

About the author

Pieter-Tjerk de Boer was born in Wildervank in 1972. In 1990, he obtained the Gymnasium B diploma at the Ichthus College in Enschede. From 1990 till 1996, he studied Technical Physics (with some additional courses from Electrical Engineering) at the University of Twente; his M.Sc. project at the Theoretical Physics group was about "Soliton switching in directional couplers". Then he started doing research at the Tele-Informatics and Open Systems group (later the Telematics Systems and Services group) at the Computer Science department of the University of Twente, resulting in this thesis. At the moment, he is working as a post-doc at the same group.

Index